Afstudeerscriptie

Spotmen het remote online spot systeem

Door P.A. van Rij

Afstudeerscriptie

Spotmen het remote online spot systeem

Afstudeerder: **Peter van Rij**
Studentnummer: **1158871**
Afstudeerbedrijf: **Postmen**
Afstudeerdocent: **Eelco Tienstra**

Voorwoord

De Mediatechnoloog heeft brede kennis van zowel media als technologie. Na een aantal jaar de focus op het mediagedeelte te hebben gehad, besloot ik dat het leerzaam zou zijn om de aandacht naar de technologie te verleggen.

Om deze scriptie tot stand te brengen heb ik stage gelopen bij postproductiebedrijf Postmen in Hilversum, Een bedrijf dat vooral in de mediahoek zit. Maar ik heb er een afstudeeropdracht aangenomen die in eerste instantie maar weinig met media te maken heeft. De heren van Postmen vroegen mij namelijk een proces binnen hun bedrijf te automatiseren door een internetapplicatie te schrijven. Software engineering binnen een mediabedrijf; mediatechnologie ten top. Ik heb hier mijn mediatechnologische kennis en vaardigheden dan ook flink uitgebreid.

De opdracht bestond uit het programmeren van een systeem, waarbij via internet videobestanden gespot kunnen worden. Dat moest gaan gebeuren in een wereld vol videoformaten, programmeertalen, EDL's, aspect ratio's, HD, h264, codecs, QuickTime, Final Cut Pro, Avid, Mac en PC, bij een bedrijf dat postproductie doet. Een mooie uitdaging.

Om dit verslag ook leesbaar te houden voor mensen zonder programmeerkennis heb ik geprobeerd het niet al te technisch te maken. Maar aangezien het programmeren het belangrijkste deel van mijn opdracht was en het bovendien deel uitmaakt van mijn opleiding, kan ik niet altijd om de techniek heen.

Wanneer ik in deze scriptie spreek van de 'klant', heb ik het over de klant van Postmen, meestal een regisseur, producent of cameraman. Postmen zelf wordt in deze scriptie ook wel opdrachtgever genoemd.

Tot slot wil ik de mensen bedanken die mijn stage mogelijk hebben gemaakt. Natuurlijk heb ik op de opleiding Mediatechnologie een brede, onmisbare basis onderwezen gekregen, maar specifiek tijdens het afstuderen heeft Hans van de Burg mij erg geholpen. Als afstudeerbegeleiders gaat mijn dank uit naar Steven van der Linden en Eelco Tienstra en als laatste wil ik vooral mijn inspirerende collega's van Postmen bedanken: Geert, Jan, Duko, Rino en Marco.

Delft, 12 April 2006

Inhoudsopgave:

	Pagina:
Voorwoord	**2**
Inhoudsopgave	**3**
1. Samenvatting	**4**
2. Inleiding	**6**
3. Achtergrond	**7**
3.1 Spotten	7
3.2 Wat is tijdcode	7
3.3 Het bedrijf Postmen	9
3.4 De Postmen workflow	10
4. Probleemstelling	**11**
5. Plan van aanpak	**12**
5.1 Bedrijfsproces	13
5.2 Research	15
5.3 Programmeren	17
5.4 Server- en internetmogelijkheden	21
5.5 Testfase	22
5.6 Implementatie	24
6. Conclusie en aanbevelingen	**25**
7. Evaluatie	**27**
Bronnen	**28**
Bijlagen	**29**
Bijlage A: EDL Breakdown	30
Bijlage B: Originele planning	31

1. Samenvatting

Achtergrond
Postmen is een breed bedrijf waar veel verschillende processen plaatsvinden. Postproductie is de hoofdmoot, maar daarbinnen bestaan veel verschillende richtingen. Bij Postmen komen klanten om gefilmd materiaal te laten monteren, te laten nabewerken, om het geluid te laten maken of bewerken of om allerlei andere nabewerkingen te laten uitvoeren. Doordat iedereen bij Postmen verschillende processen beheerst, kan Postmen voor allerlei opdrachten ingeschakeld worden.

Deze scriptie gaat over de afstudeeropdracht waarbij een bedrijfsproces van Postmen geautomatiseerd is. Het gaat dieper in op de problemen die opdoken bij dat automatiseren en op de oplossingen hiervoor. Uiteindelijk is een werkend product geïmplementeerd waardoor de *inhouse* faciliteiten van Postmen ontlast worden.

Probleemstelling
Bij het monteren van video, spotten klanten hun geschoten materiaal regelmatig bij Postmen zelf. Aangezien Postmen maar zeer beperkte faciliteiten heeft voor het spotten, is het van belang om een manier te vinden waarop klanten dat spotten bijvoorbeeld thuis kunnen doen, zodat Postmen geen klanten hoeft te weigeren wegens ruimtegebrek. De directie van Postmen wil het spotten automatiseren, zodat klanten via internet alvast kunnen spotten, zonder een beroep te hoeven doen op de spotset van Postmen.

Vraagstelling
In deze scriptie probeer ik de volgende vragen te beantwoorden:

- Welke onderdelen van het huidige spotproces kunnen geautomatiseerd worden?
- Wat zijn daarbij de wensen van Postmen?
- Wat zijn de wensen van gebruikers bij het spotten?
- Welke programmeertaal kan het best gebruikt worden?
- Welk videoformaat kan het best gebruikt worden en hoe verspreid ik deze het best via internet?

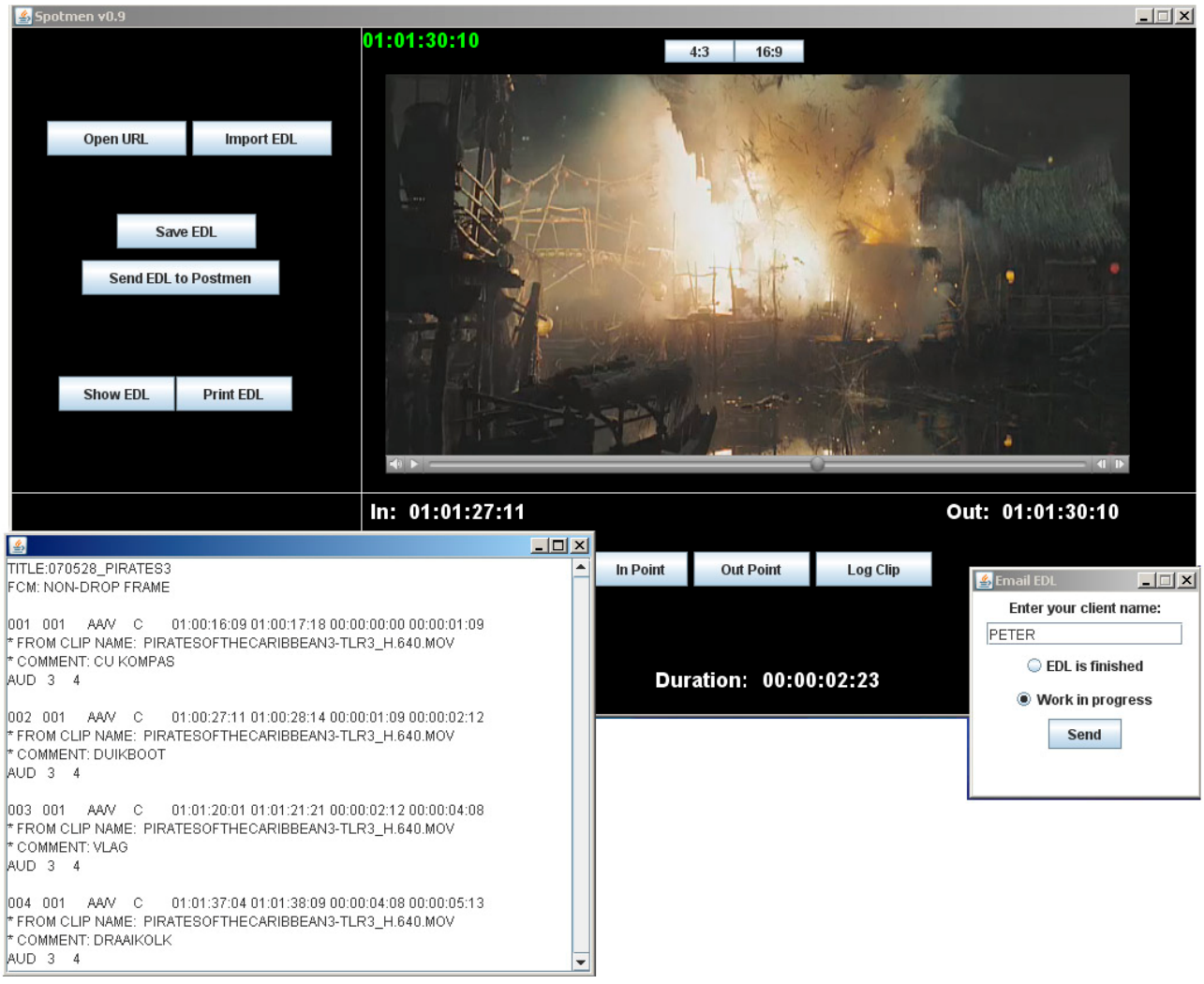
Plan van aanpak
Omdat ik vanuit het grote niets moest beginnen, heb ik een systematische aanpak gekozen voor het beantwoorden van de belangrijkste vragen. Ik begon met het leren kennen van Postmen en haar klanten. Daarna probeerde ik zoveel mogelijk te weten te komen van de verschillende mogelijkheden van verschillende programmeertalen en videoformaten. Toen moest ik gaan kiezen welk formaat en welke taal ik zou gaan gebruiken, om me vervolgens zo veel mogelijk op die twee te storten. Inmiddels moest ik ook weten hoe ik een programma en een videobestand via internet verspreidde, waarna ik alles kon testen en *finetunen*. Als laatste moest ik het geheel implementeren in de processen van Postmen. Omdat alles binnen relatief korte tijd af moest, heb ik aan het begin van het project een zo realistisch mogelijke – doch zeer strakke - planning gemaakt.

Methode
Om de afstudeeropdracht ten uitvoer te kunnen brengen heb ik voornamelijk gebruik gemaakt van internetbronnen en een aantal boeken. Ook heb ik de wensen van de klant en de opdrachtgever geïnventariseerd. Eerst heb ik uitgezocht wat de mogelijkheden zijn van de videoformaten en programmeertalen en hieruit is vervolgens een keuze gemaakt. De weg naar een succesvol programma was vol met obstakels, maar uiteindelijk zijn alle problemen opgelost, soms met behulp van de QuickTime for Java Apple mailinglist.

Resultaten
Na een onderzoek naar de mogelijkheden en verschillen van de diverse programmeertalen zoals Java, Php, C++, etc. kwam naar voren dat de beste oplossing QuickTime in combinatie met Java zou zijn. Voor Java koos ik omdat het crossplatform is: zowel Windows als Apple's Mac ondersteunen Java. QuickTime is het standaard formaat van de Mac, maar is ook voor de PC goed te gebruiken. QuickTime in combinatie met Java bleek een goede keus te zijn, want voor Java is een *native library* (bibliotheekuitbreiding) beschikbaar om de diverse interne functies van QuickTime aan te roepen. Bovendien is het QuickTime videoformaat de standaard output van het videomontageprogramma Final Cut Pro, een programma waar opdrachtgever Postmen steeds vaker mee werkt. Mede hierdoor en door de vele andere mogelijkheden die Java en QuickTime bieden, heb ik voor deze twee gekozen.

Een van de eisen was dat het Java programma door de klant via internet te benaderen moest zijn. Met Java zijn daarbij twee mogelijkheden, Java *Web Start* met een Java applicatie of een Java *applet*. Uiteindelijk is er gekozen voor de Java *Web Start* oplossing. Hierbij staat de Java applicatie op een webserver, waarop ook de content (ruw videomateriaal) staat. Via een website kan de klant zowel de applicatie als de content overal ter wereld 'tot zich nemen'.

Conclusie
De werkwijze van de klant voor het spotten van geschoten materiaal kan door middel van Spotmen (de gekozen naam voor de Java applicatie) gemakkelijker gemaakt worden. De enige benodigdheden zijn een PC met QuickTime en Java of een Mac (als de Mac up to date is, dan zijn Java en QuickTime al geïnstalleerd). Postmen zorgt dat het gefilmde materiaal op een server online staat, waarna de klant dit via Spotmen kan gaan spotten. Programmeren is niet mijn sterkste kant en dat is het nooit geweest, desondanks is het gelukt om een applicatie te schrijven die aan alle wensen van klant en opdrachtgever voldoet. Een belangrijk bedrijfsproces van Postmen is hierbij geautomatiseerd.



2. Inleiding

Deze scriptie is het resultaat van maanden onderzoek, programmeerwerk en hoofdpijn. Ik wil u graag van dat onderzoek- en programeerproces deelgenoot maken. Door een heldere verhaalstructuur te kiezen, blijft de hoofdpijn u misschien bespaard.

De scriptie is opgedeeld in grofweg twee delen. Het eerste deel beslaat het achtergrondwerk. Hierin vertel ik eerst iets over de processen waarmee ik zeer direct te maken had en leg ik voor de volledigheid ook enkele termen uit die voor mijn onderzoek van belang zijn. Vervolgens *zoom* ik in op het bedrijf Postmen en op de werkprocessen daar. Aan het eind van deel één zal ik de probleemstelling, de centrale vraag en de deelvragen uitwerken.

Deel twee kan gezien worden als een procesverslag, waarin ik in zes hoofdstukken de vorderingen beschrijf die ik vanaf dag één gemaakt heb. Die zes hoofdstukken geven de verschillende onderdelen van het project weer en zijn apart geschreven, ook al lopen zaken in het echt natuurlijk constant door elkaar. De hoofdstukken in dit tweede deel gaan achtereenvolgens over mijn **kennismaking** met het bedrijf en haar werkprocessen, over mijn **research** naar mogelijkheden die er zijn om het spotproces te automatiseren, over het **programmeren** van een internetapplicatie, over het onderzoeken van **server- en internetmogelijkheden**, over het **testen** en aanpassen van de applicatie en als laatste over de **implementatie** van het programma in het bedrijfsproces van Postmen.

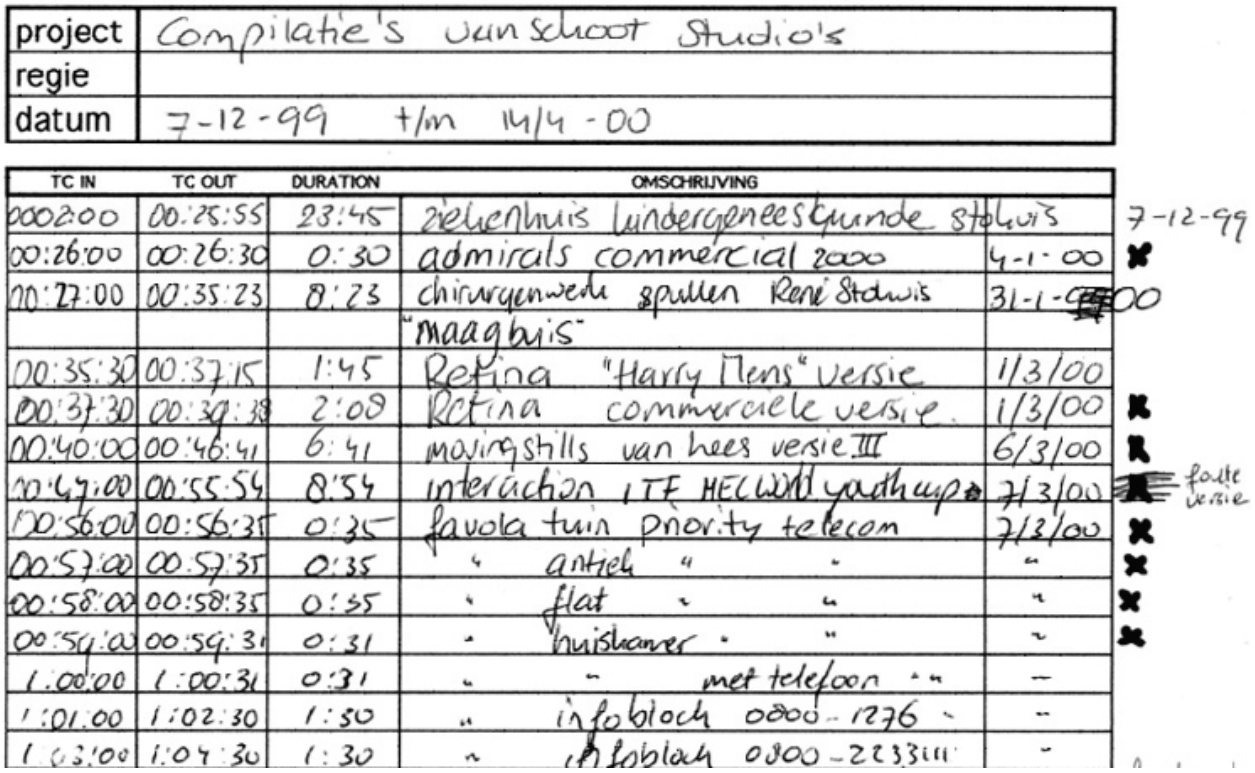
Vervolgens leest u de conclusie waarin ik de resultaten samenvat en waarin ik vertel over de toekomstige uitbreidingsmogelijkheden wat betreft het te automatiseren spotproces, waarna een korte evaluatie volgt over dit project, de stage en alles wat verder ter tafel komt.

Veel leesplezier gewenst!

3. Achtergrond

3.1 Spotten

Tijdens het monteren wordt ruw beeldmateriaal op volgorde geknipt en geplakt, zodat een film ontstaat. Omdat er vaak veel meer materiaal gefilmd wordt dan nodig, is het soms lastig om in de overdaad van materiaal de juiste beelden terug te vinden. Om die reden wordt vrijwel altijd een inventarisatie gemaakt van het gefilmde materiaal. De verantwoordelijke voor de montage, de klant, bekijkt al het materiaal en schrijft op welke beelden hij wil gaan gebruiken. Daarbij noteert hij van elk te gebruiken shot de tijdcode van het begin en einde van het fragment, hij berekent (met een speciale rekenmachine) eventueel hoe lang het fragment duurt en hij omschrijft in het kort wat er in het fragment gebeurt. Tijdens de montage kan nu elk fragment in een handomdraai teruggevonden en gemonteerd worden.
Een spotlijst is vrijwel onmisbaar bij het monteren. Omdat het een ruwe inventarisatie betreft, hoeft het allemaal niet heel precies te gebeuren, maar hoe duidelijker de spotlijst is, hoe makkelijker en sneller het monteren zal gaan.
Omdat de banden waarop professionele camera's opnemen niet in de standaard videospeler passen, is de klant afhankelijk van spotsets. Hiermee kan materiaal op een beeldscherm worden bekeken, waarbij de tijdcode van het materiaal in beeld meedraait. Op die manier kan eenvoudig meegeschreven worden en kan de klant het materiaal dat hij gefilmd heeft bovendien nog eens terugzien.



project	Compilatie's Vanschoot Studio's
regie	
datum	7-12-99 t/m 14/4-00

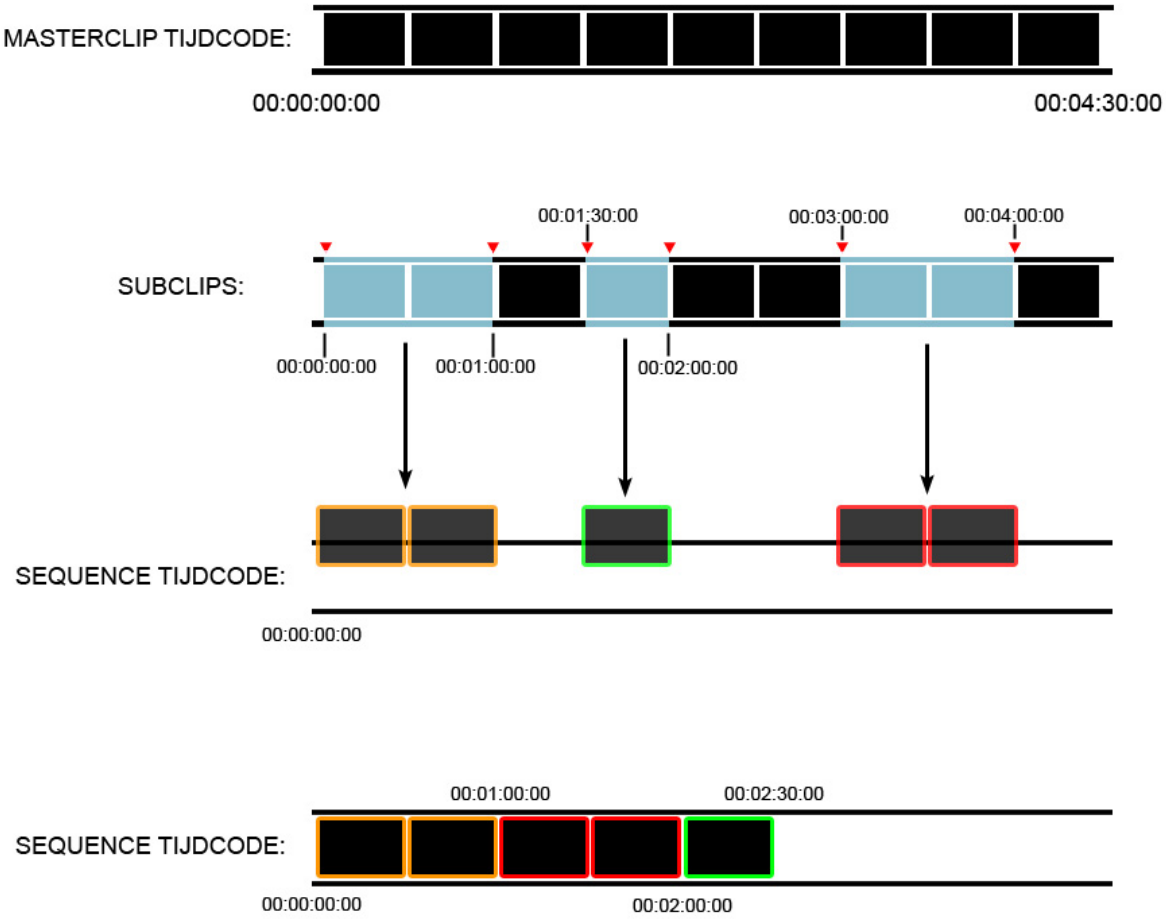
TC IN	TC OUT	DURATION	OMSCHRIJVING		
00:02:00	00:25:55	23:45	ziekenhuis Lindergeneeskunde Stolwijs		7-12-99
00:26:00	00:26:30	0:30	admirals commercial 2000	4-1-00	✖
00:27:00	00:35:23	8:23	chirurgenweek spullen René Stolwijs	31-1-00	
			"maagbuis"		
00:35:30	00:37:15	1:45	Refina "Harry Mens" versie	1/3/00	
00:37:30	00:39:38	2:08	Refina commerciele versie	1/3/00	✖
00:40:00	00:46:41	6:41	movingstills van hees versie III	6/3/00	✖
00:47:00	00:55:54	8:54	interaction ITF HEC WK youthcup	7/3/00	foute versie
00:56:00	00:56:35	0:35	favola tuin priority telecom	7/3/00	✖
00:57:00	00:57:35	0:35	" antiek " "	"	✖
00:58:00	00:58:35	0:35	" flat " "	"	✖
00:59:00	00:59:31	0:31	" huishouden " "	"	✖
1:00:00	1:00:31	0:31	" " met telefoon " "	"	
1:01:00	1:02:30	1:30	" infoblock 0800-1276 "	"	
1:03:00	1:04:30	1:30	" infoblok 0800-2233111	"	

3.2 Wat is tijdcode?

Met film- en videocamera's word altijd een tijdssignaal opgenomen tijdens het filmen. Dat is niet de huidige tijd / datum, maar een code voor hoe lang er al is opgenomen op de band. Stel men neemt beeldmateriaal op met een digitale videocamera, dan zal de tijdcode bij 00:00:00:00 beginnen, de tijdcode bestaat uit de volgende velden HH:MM:SS:FF deze velden staan achtereenvolgens voor *hours* (uren), *minutes* (minuten), *seconds* (seconden) en *frames* (beelden per seconde). De tijdcode is dus tot op het frame nauwkeurig; elk frame heeft een eigen tijdcode. Op de professionele camera's is het ook mogelijk om deze tijdcode zelf in te stellen of camera's aan elkaar te linken zodat ze met dezelfde tijdcode zijn gesynchroniseerd.
Meestal past op een videoband van een professioneler type (geen DV, maar bijvoorbeeld DigiBeta) twintig minuten beeld. Als er meerdere banden gebruikt zijn, wordt vaak een handigheidje toegepast. Bij elke gebruikte band wordt het eerste veld *hours* een uur verder gezet, 01 voor de eerste band, 02 voor de volgende en zo verder.

Als men met een lege, nieuwe band begint, staat daar standaard nog geen tijdcode op. De fout kan dan gemaakt worden dat men iets opneemt, dan terugspoelt om te kijken en vervolgens weer doorspoelt om verder te filmen. Als men bij dat doorspoelen vérder spoelt dan het einde van het vorige fragment, begint de tijdcode weer bij 00:00:00:00, omdar er nog geen tijdcode op de band stond. Om dit tegen te gaan kan men de band *zwarten* of *stripen*. Dat wil zeggen dat er onafgebroken opnamen op de band worden gemaakt voordat er écht mee gefilmd gaat worden. Meestal gebeurt dat met de dop op de lens van de camera of met een DigiBeta machine. Als er daarna opgenomen, gespoeld, opgenomen en nog meer gespoeld wordt, loopt de tijdcode altijd netjes mee.

De tijdcode is niet alleen belangrijk tijdens het spotten, maar ook bij het monteren. Montageprogramma's lezen de tijdcode mee in. Als er een beeldmontage is gemaakt, kunnen achteraf aan de hand van tijdcodes nog fragmenten opnieuw worden ingeladen. Op deze manier kan bijvoorbeeld eerst al het materiaal worden ingeladen op lage kwaliteit, dat wordt vervolgens gemonteerd en als de montage klaar is kan aan de hand van de tijdcode het materiaal nog eens op hogere kwaliteit worden ingeladen. Die laatste keer hoeft dan niet ál het materiaal ingeladen te worden, maar kan heel precies het materiaal op de computer worden gezet dat gebruikt gaat worden. Dat scheelt veel computerschijfruimte.
Als er tussen diverse montagesystemen geschakeld moet worden, wordt er ook wel gebruik gemaakt van een EDL (Edit Decision list) of AEF bestand. Dit is in weze een tekstbestand waarin de in- en uitpunten staan aangegeven van de fragmenten (clips) die gemonteerd zijn. Hierbij wordt alleen de *brontijdcode* gebruikt, de tijdcode zoals die op het originele materiaal terug te vinden is. In het EDL bestand staat van elke clip dus de brontijdcode aangegeven, alsmede de plek die de clip op de tijdlijn van het montageprogramma kreeg (de *bestemmingstijdcode*). Als de banden bewaard worden vormt het EDL bestand dus een soort back-up van de montage (het gaat hier puur om de montage, effecten, speciale beeldovergangen en kleurcorrecties staan niet in een EDL).



De *masterclip* tijdcode is de tijdcode van het bronmateriaal, bijvoorbeeld de videotapes. In deze scriptie wordt dit de brontijdcode genoemd. In dit voorbeeld is voor het gemak gekozen voor een begintijdcode van 00:00:00:00 en een eindtijdcode van 00:04:30:00. De *subclips* vormen de selectie van bruikbaar materiaal die op de tijdlijn in het montageprogramma worden gezet. Deze clips vormen achter elkaar (wellicht in een andere volgorde) de *sequence* tijdcode, die ik bestemmingstijdcode zal noemen. In EDL bestanden staat welke clips waar staan op de tijdlijn (welke bestemming ze hebben), maar staat ook wat de brontijdcode (*masterclip*, de tapes) van de clips is.

3.3 Het bedrijf Postmen

Postmen is een postproductiebedrijf in de meest brede zin van het woord. Geert van Schoot en Jan de Wit, grondleggers van het bedrijf, begonnen ooit met alleen audiopostproductie, maar inmiddels hebben ze ook de kennis, apparatuur en opdrachten in huis gehaald voor videopostproductie, videomontage, 3D animatie, DVD *authoring* en nog vele andere zaken.

Het voordeel van Postmen is dat ze alles *inhouse* kunnen doen, waardoor het voor de klant eenvoudiger en voordeliger is om een product te realiseren. In de omgeving van Hilversum is deze brede benadering vrij uniek. Veel bedrijven zijn juist gespecialiseerd in één enkel proces binnen de nabewerking, vaak ook nog eens alleen in beeld of geluid.

Postmen kan klanten het hele pakket bieden omdat het personeel heeft dat van alle markten thuis is. Iedereen heeft een eigen specialiteit maar heeft tegelijkertijd verstand van meerdere softwarepakketten uit de business. Wat dat betreft is Postmen een geschikt bedrijf voor de brede kennis en opleiding van een mediatechnoloog.

Om een indruk te geven van wat Postmen allemaal doet, hieronder een paar projecten en een korte omschrijving van het aandeel van Postmen daarin.

Leader Knoop in je zakdoek
Voor het programma ‘Knoop in je zakdoek’ heeft Postmen de gehele vormgeving, concept, uitwerking en audiovormgeving verzorgd van de nieuwe leader en ook decorstukken voor in de show. De leader is met *stop motion* techniek gemaakt, door foto's te maken met een digitale camera en deze vervolgens als een animatie achter elkaar te zetten. De nabewerking van de leader, de montage en *compositing* zijn met behulp van Adobe After Effects 7 gedaan.

Ajax jeugdtraining DVD serie
Dit ambitieuze project bestaande uit acht dvd's is gemonteerd door Duko met een Avid, waarna hij tevens de nabewerking heeft gedaan met Adobe After Effects 7. De 3D animaties zijn afkomstig van Nazooka, een Belgisch bedrijf. Deze animaties zijn via ftp bij Postmen gedownload en in de montage toegevoegd. De dvd serie wordt wereldwijd verspreid, de releasedatum is op het moment van schrijven nog niet bekend.

Najib Amhali’s Most Wanted
Van deze populaire show van Najib Amhali, die bij de uitzending op televisie door 2 miljoen mensen werd bekeken, is een DVD gemaakt. Postmen nam de montage en postproductie van deze DVD op zich. Geert deed de videomontage en samen met Marco deed hij de audio postproductie. Vervolgens ontwierpen Duko en Rino het DVD menu met behulp van DVD Studio Pro. De DVD *authoring* is tenslotte door Jan gedaan.

Hans Klok filmpjes
Om zijn magische illusies in te leiden, wilde de illusionist Hans Klok gebruik maken van filmpjes. Door deze filmpjes aan het publiek te tonen, kon er ook een nieuwe illusie en decor worden opgebouwed. Er zijn in totaal drie filmpjes gemaakt, ieder met een ander onderwerp. De hele productie is door Postmen verzorgd, van scenario schrijven tot produceren en regisseren. Rino nam de rol van regisseur op zich. Het materiaal is gefilmd met een Panasonic camera op HD kwaliteit. De nabewerking gebeurde later in After Effects en 3D Studio Max.

3.4 De Postmen Workflow

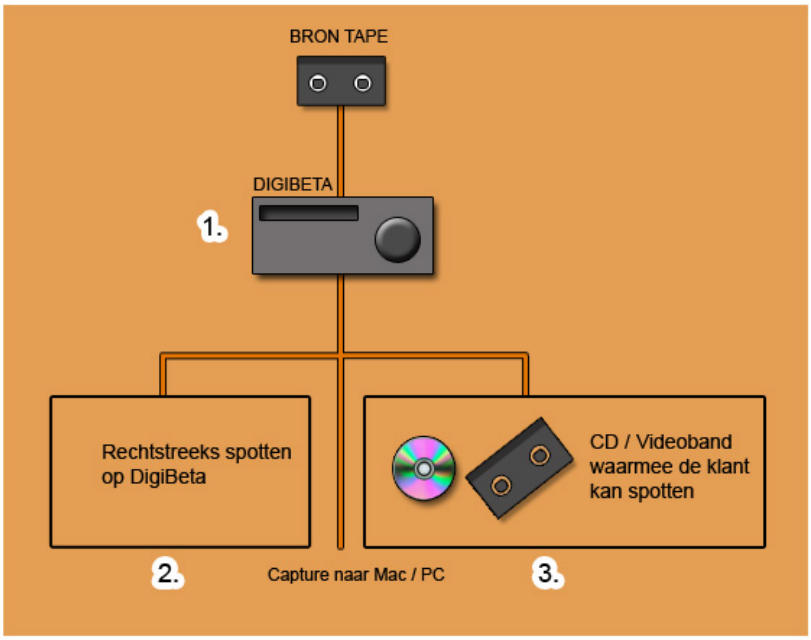
Als klanten willen spotten kunnen ze gebruik maken van de Postmen faciliteiten. Toen ik aan mijn afstudeeropdracht begon waren er twee methodes om te kunnen spotten. Postmen heeft een DigiBetamachine waar de professionele banden in passen. Daarop is een scherm aangesloten waarop het materiaal gekeken kan worden. De klant reserveert de machinekamer bij Postmen. Op het afgesproken tijdstip komt de klant langs bij Postmen en gaat in de machinekamer zitten. De klant start vervolgens de band met gefilmd materiaal waarop een tijdcode meeloopt in beeld. De klant kan op een lijst de shots opschrijven, oftewel: spotten. Als er gemonteerd wordt, houdt de klant deze lijst bij de hand.

Omdat er maar één DigiBeta bij Postmen aanwezig is én omdat de klant hiervoor naar Postmen toe moet komen, wordt ook wel een andere, meer omslachtige methode toegepast. Postmen krijgt (bijvoorbeeld per post of per koerier) banden aangeleverd. Een medewerker stopt de band in de DigiBeta speler en kopieert het materiaal op analoge snelheid naar een VHS videoband of DVD – de *zichtkopie*. Op deze zichtkopie wordt de brontijdcode mee gekopieerd (ingebakken), zodat je die tijdcode tijdens het afspelen altijd in beeld ziet. Deze zichtkopie wordt vervolgens weer naar de klant gestuurd zodat die thuis of op een andere locatie kan spotten. Omdat de band op analoge snelheid gekopieerd wordt en omdat er maar één DigiBeta is, kan er tijdens het maken van een zichtkopie ook enige tijd niet gespot worden. De klant kan dus op twee manieren spotten: voor de ene moet hij naar Postmen toe en voor de andere moet Postmen kopieën maken en die naar de klant sturen. Voor beide gevallen geldt dat er altijd maar één klant tegelijkertijd geholpen kan worden.

In het afstudeerproject moet alles gedigitaliseerd en geautomatiseerd worden. Als dat gelukt is maakt het spotten in zekere zin deel uit van het montageproces. Als banden gedigitaliseerd worden voor montage gebeurt dat namelijk ook met die ene DigiBetamachine. Hieronder staat uitgebeeld hoe dat proces van inladen van beeldmateriaal ongeveer werkt. Dit proces dient als één van de peilers voor de probleemstelling van deze scriptie: *Hoe kan Postmen het bedrijfsproces van het spotten automatiseren, zodat klanten dat thuis of op kantoor kunnen doen en zodat Postmen haar tijd en middelen efficiënter kan besteden?*

Mocht de klant na het spotten de montage bij Postmen willen doen, dan vinden de volgende stappen plaats. De tape wordt door Postmen gedigitaliseerd naar een Avid of Final Cut Pro set, via bijvoorbeeld de DigiBetamachine of met behulp van een camera die tijdelijk wordt aangesloten op de montageset. In de montagesoftware is vervolgens de brontijdcode van het bronmateriaal zichtbaar te maken. Aan de hand van de spotlijst van de klant kan het ingeladen beeldmateriaal vervolgens gemonteerd worden.

SPOT WORKFLOW V/D KLANT



1. Tape wordt in DigiBetamachine gestopt.
2. Klant kan rechtstreeks de bron tape spotten.
3. Van de source tape kan een kopie gemaakt worden naar DVD of videoband met tijdcode in beeld .

4. Probleemstelling

Producties voor de televisie of bedrijfsfilms hebben vaak een strakke deadline. Daarom is het belangrijk om de workflow zo efficiënt mogelijk te laten verlopen. Op dit moment levert dat soms nog problemen op. Het gefilmde materiaal dat de klant op band heeft staan moet worden ingeladen, maar het materiaal moet ook gespot worden. De volgende dag is de montage vaak al gepland. Zo gebeurt het dat de klant zonder uitgebreide spotlijst aan de montagetafel verschijnt, wat weer extra montagetijd tot gevolg heeft.

Het kost relatief veel tijd om de klant bij Postmen te laten spotten en daarna het materiaal in te laden. Uiteraard levert dit geld op, maar een sneller en efficiënter systeem is gewenst. Daarmee biedt je de klant ook extra service. Die mag immers kiezen wat hij doet, hij blijft te allen tijde welkom om gewoon bij Postmen zelf te spotten.

De tendens in de tv- en videowereld is dat veel zaken die vroeger gedaan werden door specialistische bedrijven, met de huidige techniek heel goed zelf thuis door iedereen zijn uit te voeren.

Denk bijvoorbeeld aan het monteren zelf: een regisseur kan op zijn laptop al een heel aardige video- of audiomontage maken, zonder dat daarbij een professionele studio aan te pas hoeft te komen. Je zou kunnen denken dat dit een nadeel is voor de productiebedrijven, maar als een klant eerst thuis een voormontage maakt, kan Postmen weer extra tijd aan andere klanten besteden. Voor de hoofdmontage of montage van hoge kwaliteit beeldmateriaal wordt op dit moment nog ruimschoots studiotijd ingekocht.

Die combinatie van thuis kunnen werken en toch een studio nodig hebben, biedt voor het spotten nieuwe perspectieven. Als de klant via internet vast een uitgebreide spotlijst kan maken voordat hij daadwerkelijk achter de montagetafel plaatsneemt, scheelt dat voor iedereen tijd (en geld). Met dat beeld voor ogen, plus alle kennis over het spotten in het achterhoofd, kom je tot de conclusie dat de Postmen spotworkflow er idealiter als volgt uit zou moeten zien.

Op het moment dat Postmen een tape inlaadt wordt er gelijk een lage resolutie (lowres) proxy gegenereerd – een tijdelijk bestand. Deze lowres versie moet vervolgens snel op een server terecht komen. Op de server of website van Postmen is een aparte klantenlogin aanwezig waar de klant onder zijn eigen unieke naam kan inloggen. In dit aparte deel van de website zal dan een applicatie draaien waarmee de klant een videofile streaming kan bekijken en tegelijkertijd ook al kan spotten. In de QuickTime file die dan gebruikt gaat worden is ook nog de originele tijdcode aanwezig als een embedded QuickTime track. Als de klant klaar is met spotten, moet er een EDL file gegenereerd worden. De klant kan dit bestand via de applicatie naar Postmen sturen. Postmen kan deze EDL inladen in Final Cut Pro en vervolgens het gewenste videomateriaal inladen. Ook kan met dit bestand een bestaande montage aangepast worden of kunnen bronbestanden (masterclips) opnieuw ingeladen worden.

Vraagstelling
In bovenstaande toekomstvisie verraad ik al een beetje welke richting ik heb gekozen, namelijk de keuze voor het QuickTime videoformaat. Dat was een eenvoudige keuze waar u later meer over zult lezen. Andere keuzes vergden méér onderzoek voordat er conclusies konden worden getrokken. Aan dat onderzoek gingen enkele vragen vooraf. Voor het grootste deel zijn die vragen uit de toekomstvisie te extraheren, met als belangrijkste de vraag hoe die toekomstvisie gerealiseerd zou kunnen worden. Dat is indirect dan ook de ***hoofdvraag:***

- Hoe kan Postmen het bedrijfsproces van het spotten automatiseren, zodat klanten dat thuis of op kantoor kunnen doen en zodat Postmen haar tijd en middelen efficiënter kan besteden?

Omdat deze vraag te veelomvattend is om direct te kunnen beantwoorden, heb ik hem opgesplitst in ***deelvragen:***

- Wat zijn de wensen van Postmen bij het automatiseren van het spotproces?
- Wat zijn de wensen van de klant bij het internetspotten?
- Met welke programmeertaal kan ik het best werken als ik het spotten wil automatiseren?
- Hoe kan ik een spotapplicatie het best via internet verspreiden?

Door mijn scriptie heen probeer ik antwoord te geven op de deelvragen en, hopelijk, daarmee ook op de hoofdvraag. Daarnaast beschrijf ik hoe ik die vragen heb proberen te beantwoorden door ook daadwerkelijk een applicatie te schrijven waarmee klanten via internet kunnen spotten.

5. Plan van aanpak

Alles tegelijk. Dat is ongeveer mijn werkmethode geweest bij dit afstudeerproject. Toch heb ik dit ‘procesverslag’, waarin ik antwoord geef op de gestelde hoofd- en deelvragen, opgedeeld in zes logische hoofdstukken, alsof ik ze tijdens het project netjes stukje voor stukje heb afgehandeld. In het echt liep alles door elkaar, kon het één niet zonder het ander en was ik soms net zo makkelijk met het tweede als met het laatste deel tegelijk bezig. Toch zijn er logische fases in de opdracht te onderscheiden. Bovendien zou ik u de hoofdpijn besparen, onderstaande indeling leek me dus de meest wenselijke.

5.1 Bedrijfsproces
Als eerste was het belangrijk om de huidige werkwijze van Postmen in kaart te brengen zodat er duidelijk werd wat de werkwijze van de klanten en medewerkers bij Postmen was.

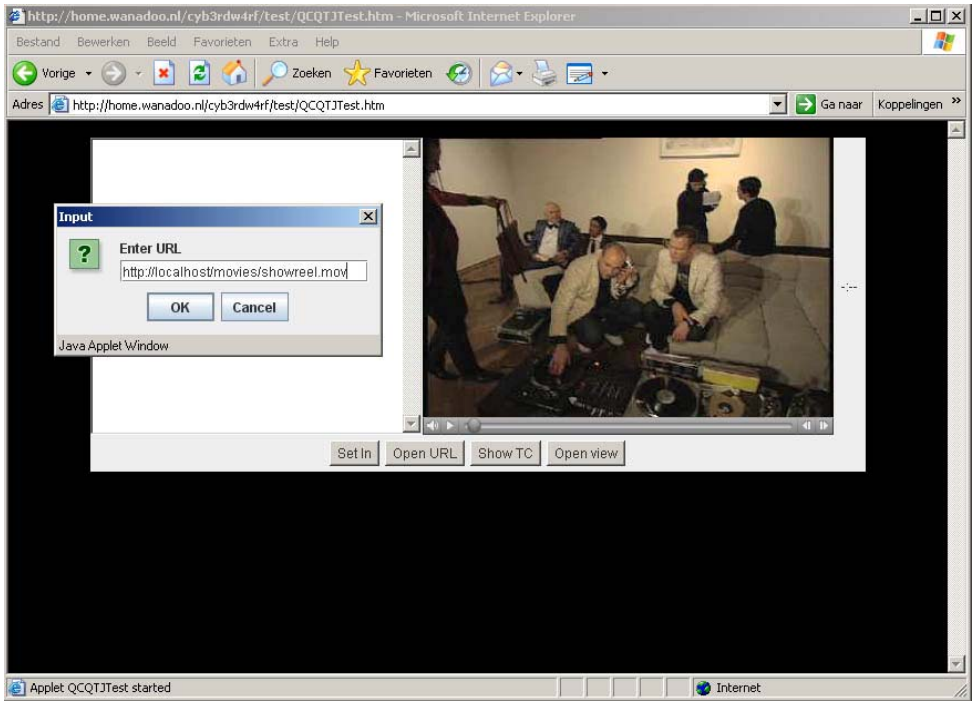
5.2 Research
Nadat ik de workflow in kaart had gebracht was de volgende stap om via internet research te doen naar de mogelijkheden van videoarchitecturen, EDL bestanden, videomontage software en programmeertalen.

5.3 Programmeren
Nadat een geschikte programmeertaal in combinatie met videoarchitectuur was gevonden is het grootste deel van de afstudeertijd besteed aan het leren van de programmeertaal Java.

5.4 Server- en internetmogelijkheden
Spotmen moest via internet kunnen draaien, dus moest er ook een geschikte manier bepaald worden om de Java applicatie en de QuickTime movies voor de klanten beschikbaar te maken.

5.5 Testfase
Na het leren van de programmeertaal en het onderzoeken van een geschikte server moest er uiteindelijk een werkend programma zijn. Nadat een eerste versie van Spotmen geschreven was, moest deze getest worden op verschillende systemen en verbeterd worden.

5.6 Implementatie
Als laatste was het belangrijk om te beschrijven wat de klant moest doen om Spotmen te kunnen draaien en welke stappen Postmen nog moet ondernemen om Spotmen operationeel te maken.



Een test versie van een applet om QuickTime for Java functies te testen.

5.1 Bedrijfsproces

Om mij zo goed mogelijk in te leven in de klanten die Spotmen moeten gaan gebruiken heb ik naast het programmeren ook andere taken bij Postmen gedaan en diverse zaken uitgezocht. Mede door mijn probleemoplossend gericht zoeken en denken, wat misschien wel typisch werk is voor een mediatechnoloog, kon ik al gauw met oplossingen komen. Zo heb ik heel wat uitzoekwerk gedaan over *Matchmoving*; Postmen wilde namelijk weten wat daarvan de mogelijkheden zijn en wat de werkwijze is.

Matchmoving komt van pas als er aan *live action* gefilmd materiaal (met bewegende camera) computer gegenereerde elementen toegevoegd moeten worden. Bij *Matchmoving*, of *cameramatching* genereer je een virtuele camera aan de hand van gefilmd materiaal, waarbij ook *photogrammetry* en projectietechnieken een rol spelen. Ik heb meerdere *Matchmoving* software onderzocht zoals *Boujou*, *Matchmover Pro* en *Syntheyes*. Verder heb ik een boek (*Matchmoving*, *The invisible art of camera tracking*) gelezen, een aantal artikelen bestudeerd en heel wat internet research gedaan. Achteraf kon ik Postmen adviseren over de mogelijkheden van Matchmoving, over de te volgen werkwijze bij het filmen van materiaal en over de bestaande softwarepakketten die *Matchmoving* mogelijk maken.

Ook voor de afwisseling van het programmeren heb ik onderzoek gedaan naar *compositing*, bijvoorbeeld naar softwarepakketten als *Digital Fusion*, *Combustion*, *After Effects*, *Shake* en *Photoshop*. Dat was ook een manier om het bedrijf beter te leren kennen.

Aan het begin van de afstudeerperiode heb ik twee filmpjes gemonteerd, één voor Radio 538 en een promotiefilmpje voor een nieuw kledingmerk.

Postmen doet niet alleen maar postproductie, maar voert ook zelf producties uit waarbij werken op locatie geen uitzondering is. Zo ben ik een keer in aanraking gekomen met *chromakey*werk (*greenscreen*) en een andere keer met *stopmotion*, een methode waarbij opnames gemaakt worden met een fotocamera.

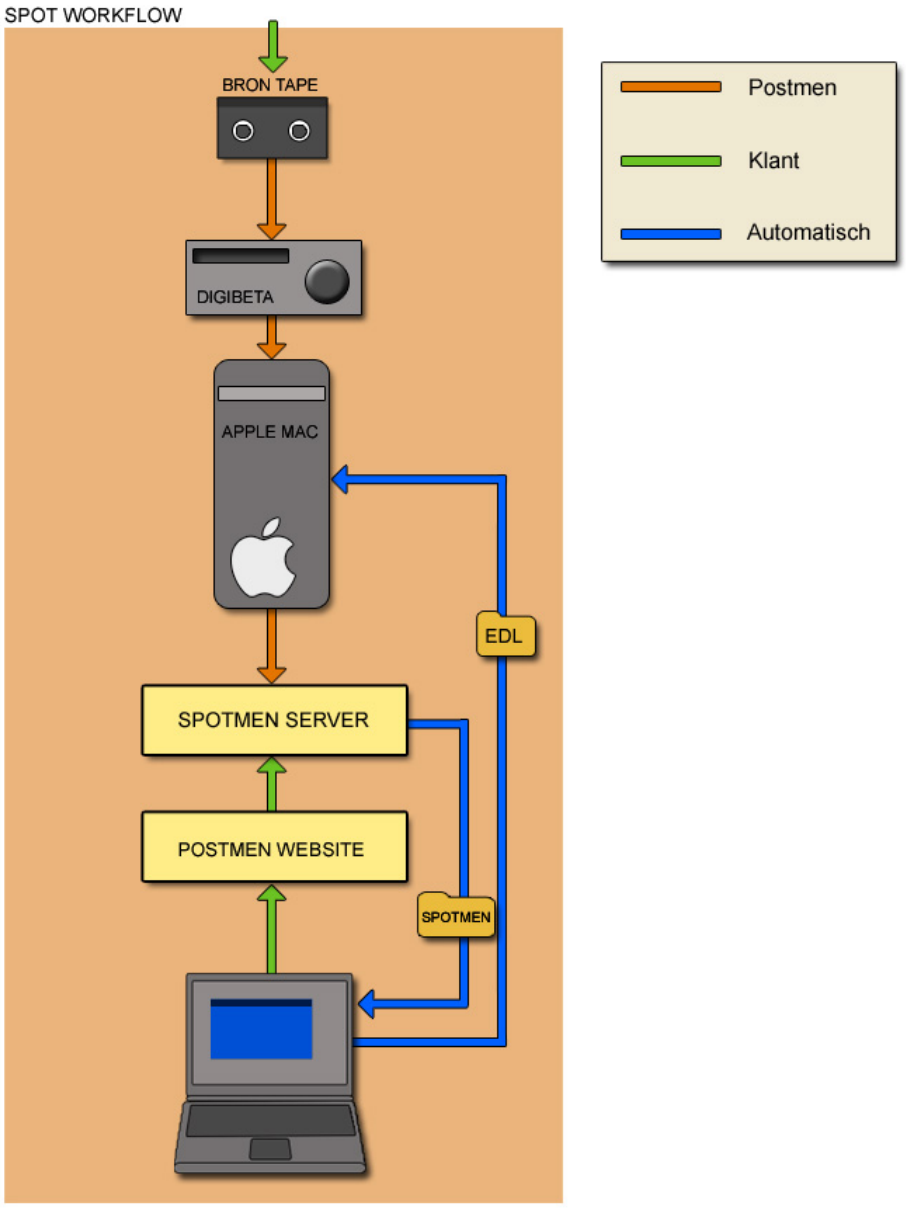
Vanaf dag één speelde door mijn hoofd de vraag hoe ik de opdracht voor Postmen moest gaan volbrengen. Tijdens de vele contacten met klanten, collega's en de opdrachtgevers hebben we vaak over die vraag nagedacht: wat is de beste manier om het spotproces te automatiseren. Sommige zaken waren vanzelfsprekend, zo moet het programma natuurlijk zo gebruiksvriendelijk mogelijk zijn. Hieronder staat in het kort wat de belangrijkste eisen voor het online spotsysteem waren.

- Postmen wil vanuit een gecaptured videobestand snel een lage resolutie versie kunnen genereren, zowel uit Final Cut Pro als uit Avid.
- Postmen wil eenvoudig videobestanden online kunnen zetten.
- Klanten van Postmen zouden via een website moeten kunnen inloggen, waarna ze bij hun eigen – sterk gecomprimeerde – videobestand kunnen komen. Op die manier hoeven klanten niet bang te zijn dat een ander hun materiaal kan zien.
- De klant moet de mogelijkheid hebben om het videobestand te spotten; dus in- en uitpunten neerzetten, een korte beschrijving geven en een eventuele waardeaanduiding van de gespotte shots geven. Dit alles moet worden opgeslagen in een tekstbestand.
- Als de klant klaar is met het spotten moet er een EDL file gegenereerd worden dat vervolgens bij Postmen weer ingelezen kan worden.
- De applicatie moet crossplatform werken zowel voor klanten met een Mac als klanten met een PC.

Heel praktisch betekent dit ook dat de dagelijkse werkwijze van Postmen en haar klanten zou kunnen veranderen. Dat is ten slotte ook het doel van het systeem. Op de volgende pagina staat hoe de workflow zoals eerder besproken in *De Postmen workflow* er uit zou moeten gaan zien, nadat de spotapplicatie Spotmen geïmplementeerd is.

Gewenste en onderzochte werkwijze na automatisering:

- De klant komt met een tape aan bij Postmen (of de tape komt per koerier);
- Bij Postmen is een medewerker die deze tape digitaliseert op de gewenste kwaliteit. Dit gebeurt bij voorkeur met Final Cut Pro;
- Via de mediamanager van Final Cut Pro wordt een *proxy lowres* versie aangemaakt in het QuickTime formaat;
- Dit QuickTime bestand wordt vervolgens geupload naar een server;
- De klant kan thuis via internet/Java Web Start de Spotmen Java applicatie starten;
- Via deze applicatie logt de klant in met zijn/haar gegevens;
- Vervolgens kiest de klant een QuickTime movie (klant ziet alleen zijn eigen movie bestanden);
- De klant spot via de Java applicatie het gekozen filmbestand;
- Na het spotten wordt een EDL bestand gegenereerd.
- De klant kan na het spotten het EDL bestand op zijn harde schijf opslaan;
- Klant geeft na het spotten de status ‘klaar’ of ‘nog niet klaar’ aan in de Spotmen applicatie waarna de EDL met de bijbehorende status naar Postmen wordt gestuurd;
- Postmen ontvangt in een door Spotmen gegenereerde email van de klant het EDL bestand;
- Postmen importeert het EDL bestand in Final Cut Pro;
- Postmen linkt het bronmateriaal aan het EDL bestand;
- De klant komt bij Postmen en constateert verheugt dat alle shots gespot en gesneden in Final Cut Pro staan en dat het werkelijke monteren direct kan beginnen.



5.2 Research

Postmen werkt met twee hoofdvideomontagesystemen: Avid en Final Cut Pro 5. De laatste tijd wordt er steeds meer met Final Cut Pro gedaan. Het bedrijf heeft ook meer Final Cut Pro montagesets. Hierdoor is de keus gemaakt om het Spotmen systeem / programma op Final Cut Pro in te richten. Final Cut Pro werkt met het QuickTime formaat.

In het originele idee van de toekomstvisie zou er automatisch een lage kwaliteit videobestand moeten worden gegenereerd van een op hoge kwaliteit *gecaptured* video bestand. Dit valt helaas op dit moment (nog) niet te automatiseren in Final Cut Pro, hierdoor is dit onderdeel van automatisch omzetten komen te vervallen. Final Cut Pro ondersteunt wel het XML formaat, wat voor de toekomst zeker nieuwe perspectieven en mogelijkheden biedt.

Er zijn diverse programmeertalen die met QuickTime kunnen communiceren, maar er zijn maar weinig programmeertalen waarmee je zowel de Mac als de PC kunt bedienen. Om de meest basale dingen voor elkaar te krijgen kan er in een html-bestand wat extra code worden ingevoegd die een aantal parameters van de QuickTime movie kan aanroepen. Daarmee kom je echter niet verder dan commando's als *play*, *loop*, de verhoudingen aanpassen of de *controlbar* van QuickTime tonen of verbergen.

Hieronder is een voorbeeld te zien van de basis QuickTime functies geïntegreerd in een html website.

```
<html>
<head>
<title>Simple QuickTime page</title>
</head>
<body>
<h2>Simple use of &lt;embed&gt; or &lt;object&gt; to put
a QuickTime movie on a web page</h2>

<!-- note that the height is 16 pixels greater than the movie's
     real height, to accomodate the control bar -->
<!-- surround the embed tag (most browsers) with the object
     tag used by IE (because they're special) -->

  <object classid="clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B"
     width="160" height="136"
     codebase="http://www.apple.com/qtactivex/qtplugin.cab">
       <param name="src" VALUE="buhbuhbuh.mov"/>
       <param name="autoplay" VALUE="true"/>
       <param name="loop" VALUE="true"/>
       <param name="controller" value="true"/>
  <embed src="buhbuhbuh.mov" width="160" height="136"
      scale="tofit"
      controller="true"
      autoplay="true"
      loop="true"
      pluginspage="http://www.apple.com/quicktime/download/"/>

  </object>


</body>
</html>
```

De eisen van de afstudeeropdracht waren natuurlijk vele malen hoger dan bovenstaande basisfuncties. Om QuickTime uitgebreider te kunnen aansturen en te programmeren moest de QuickTime API (*Application Programming Interface*, dit is een verzameling definities op basis waarvan een computerprogramma kan communiceren met andere programma's) kunnen worden aangesproken. Via de Apple Developer website (QuickTime is van Apple) werd duidelijk dat er een aantal oplossingen waren. De QuickTime API is bedoeld voor procedural C of C++ programmeurs maar het is ook mogelijk om middels Java via een *native library* (de QuickTime for Java library) de QuickTime API te gebruiken.

Omdat het nieuwe spotsysteem Spotmen crossplatform moest worden (zowel voor de Mac als de PC klanten te gebruiken) vielen eigenlijk alle programmeertalen af behalve Java en Flash *actionscript*. Flash zou met het Flash videoformaat moeten werken, waarvoor je de QuickTime movies rechtstreeks om zou moeten zetten. Met QuickTime for Java hoefde de QuickTime movie niet omgezet te worden. Doordat de programmeertaal Java als rode draad door de opleiding mediatechnologie loopt en er een QuickTime for Java library bestaat had ik al snel de keus voor Java gemaakt. Java is crossplatform en heeft mogelijkheden om programma's via internetsites uit te voeren. QuickTime is een videoarchitectuur die zowel op de PC als de Mac gebruikt wordt; uitermate geschikt voor een crossplatform programma.

Na het bestuderen van de QuickTime for Java API werd duidelijk dat deze combinatie van programmeertaal en videoarchitectuur alle mogelijkheden biedt die nodig zijn voor het Spotmen programma.

De combinatie heeft echter ook een nadeel: over QuickTime for Java is over het algemeen weinig informatie te vinden. De informatie die te vinden is beperkt zicht tot de volgende bronnen:

- Websites: Via de website van Apple is veel algemene QuickTime informatie te vinden. Verder kun je via de Apple Developers site, de 'QuickTime for Java API' en wat voorbeeldprogramma's halen. Via Google kun je een oude, uitgebreide tutorial van QuickTime for Java vinden, maar deze is te gedateerd om met de huidige classes van de QuickTime for Java library een goed werkend programma te genereren.

- Literatuur: Er is welgeteld één boek te koop over Quicktime for Java, dat heb ik dan ook meteen via amazon.com aan mijn studiemateriaal toegevoegd. Dit boek bevat gelukkig talloze tutorials. De codevoorbeelden waren alleen niet in de meest eenvoudige Java vorm geschreven. Nadat ik meer kennis kreeg van de Java taal was het boek beter te begrijpen. De Java code uit het boek kon ik later ook goed gebruiken voor het Spotmen programma.

- Mailinglist: De QuickTime for Java mailinglist bleek uiteindelijk een belangrijke bron van kennis. Het probleem is echter dat de lijst niet erg actief is. Toch kwam hij goed van pas bij het technische vraagstuk over het uitlezen van de tijdcode track van een QuickTime movie (zie: *Programmeren*).

Nadat ik de keus voor Java had gemaakt heb ik mezelf systematisch opnieuw leren programmeren in deze taal door boeken te lezen en bijbehorende programma's simpelweg over te typen. Dit waren ongeveer tweehonderd verschillende programma en hierdoor werd ik snel bekend met de Java *syntax*. Een aantal codevoorbeelden uit het QuickTime for Java Notebook heb ik ook uitgewerkt om bekend te raken met de QuickTime for Java library. Aangezien deze library is omgezet vanuit de originele C++ code wijkt hij enigszins af van de gangbare Java classes code.

Een van de belangrijkste zaken die het Spotmen programma moest kunnen was om de brontijdcode track die aanwezig is in de QuickTime movie te laten zien en te gebruiken. Als er met Final Cut Pro beeldmateriaal is ingeladen met een brontijdcode dan zit deze tijdcode automatisch in de gecapturede QuickTime movie. Er is alleen geen enkele mogelijkheid om met de QuickTime player deze brontijdcode te tonen om op die manier te kijken of de brontijdcode ook wel echt aanwezig is. Bij de filmkenmerken van de QuickTime movie in de QuickTime player staat wel dat er een tijdcode track aanwezig is, maar deze staat op een grootte van nul kilobytes waardoor het lijkt alsof er dus geen inhoud in de tijdcode track zit. Bovendien kan die track niet getoond worden in de player. Lange tijd wist ik niet of het überhaupt mogelijk was om een tijdcode uit te lezen, maar in de QuickTime API kwam ik steeds classes tegen die begonnen met 'TimeCode', dus ik hield goede hoop. Een C++ programma'tje van de Apple Developers site maakte uiteindelijk de QuickTime movie tijdcode track zichtbaar. Met dit programma kon tijdcode worden toegevoegd aan het filmbestand, of de brontijdcode kon ermee zichtbaar gemaakt worden in de QuickTime player.

Om de QuickTime movie voor de klant beschikbaar te stellen via Spotmen waren er drie mogelijkheden: Het downloaden van de QuickTime movie, het streamen van de QuickTime movie of gebruik maken van een progressive download. Het downloaden van de QuickTime movie wilde ik voorkomen in verband met copyright van beeldmateriaal. Streaming video heeft nog een groter probleem: deze variant maakt het spotten vrijwel onmogelijk. Een klant moet tijdens het spotten willekeurig door het filmpje heen kunnen scrollen, zonder dat de video steeds opnieuw geladen moet worden in een buffer op de computer. De keus viel dus op de progressive download, die deze problemen niet kent. Met deze methode wordt het filmpje in een tijdelijk bestand gedownload. Zodra de eerste bytes binnen zijn, kun je het bestand direct afspelen, terwijl het filmbestand ondertussen verder gedownload wordt. Een snelle werkmethode dus.

Bij gebruik van de progressive download moet wel alle belangrijke informatie, zoals de video- en audiocodec van het filmbestand, aan het begin van dat bestand staan (in de header). Als die informatie pas aan het einde staat, moet het filmpje toch eerst helemaal worden gedownload, waardoor de snelle werkmethode weer vervalt. Dat probleem ondervond ik zelf ook, omdat Final Cut Pro bij het maken van lage resolutie QuickTime movies die informatie aan het eind zet. Ook hier heb ik gelukkig een oplossing voor gevonden: als je het lage resolutie bestand uit Final Cut Pro opent in de QuickTime player en vervolgens opslaat als een *self-contained* QuickTime movie, verplaatst QuickTime de informatie weer naar het begin zonder verder iets aan het filmpje te veranderen. Zo hoeft de klant niet eerst te wachten totdat het volledige bestand gedownload is.

5.3 Programmeren

De meeste tijd van de afstudeerstage ging naar het programmeren: het leren van de programmeertaal, problemen oplossen, testen en *debuggen*. Hieronder staan alle belangrijke dingen op een rijtje, daarna ga ik wat dieper in op enkele specifieke problemen waar ik tijdens het programmeren tegenaan liep.

De volgende dingen moest ik doen:
- Een keuzelijst in Java maken, waarin klanten na het inloggen bijvoorbeeld kunnen kiezen met welk filmbestand ze willen gaan werken.
- Toetsenbord events gebruiken in Java, zodat je met het toetsenbord QuickTime aan kunt sturen.
- Bestanden in- en exporteren, bijvoorbeeld EDL-bestanden.
- Tekst e-mailen vanuit de Java applicatie naar een e-mail adres.
- Het Java programma opdelen in verschillende methodes.
- De verhouding van een QuickTime movie aanpassen van 4:3 naar 16:9 en andersom.
- Een venster maken waarin je met de muis kunt scrollen.
- Tijdcodes van een QuickTime movie uitlezen.
- Met tijdcodes rekenen, deze van elkaar aftrekken.
- Frames (beelden per seconde) omzetten naar een tijdcode.
- Tijdcode omzetten naar Strings in de Java programmeertaal.
- Strings via programmeercode in een tekstveld opzoeken en vervangen voor een andere string.
- Een printfunctie inbouwen zodat klanten de EDL kunnen uitprinten.
- De naam van een QuickTime movie uit een url filteren.
- Een QuickTime movie aan een Java Frame toevoegen.
- Een goede layout maken met de Java Layout managers.
- Een url inladen in een QuickTime window.
- De focus leggen op panelen zodat de toetsenbord events op het goede paneel van toepassing zijn.
- De afspeelsnelheid van een QuickTime movie aanpassen.
- De huidige datum opvragen en in Java gebruiken.

Na de research ontstonden al snel de problemen. Moest ik voor Spotmen gebruik maken van een Java applet of van een Java applicatie? In eerste instantie was nog niet duidelijk welk systeem de voorkeur had en ook de mogelijkheden moest ik nog uitzoeken. Hierdoor moest ik mij zowel verdiepen in het applet programmeren als in het applicatie programmeren. In het boek over QuickTime for Java worden de codevoorbeelden geschreven als Java applicaties, maar om makkelijk te kunnen testen op de Mac en PC was het handig om een applet op een website te zetten, hiervoor moest de applicatie code wel eerst herschreven worden in applet vorm.

Aan het begin was ik nog niet bezig met het uiteindelijke programma, eerst was het belangrijk dat ik de Java taal en vooral de QuickTime for Java mogelijkheden leerde kennen.

Met de 'QuickTime for Java' classes kan QuickTime op een hoog niveau aangesproken worden, zodat de *embedded* tijdcode track kan worden uitgelezen. Maar met deze QuickTime for Java classes kun je bijvoorbeeld ook QuickTime bestanden afspelen, exporters maken, de afspeelsnelheid van een QuickTime movie aanpassen, montage programma's maken, een videocapture systeem maken, iTunes informatie uitlezen, tijdcode tracks toevoegen, effecten toevoegen, etc. Eigenlijk kun je met deze class alles doen wat normaal mogelijk is met QuickTime *en nog veel meer*.

Terwijl ik nog losse onderdelen aan het testen was ben ik ook begonnen met het definitieve programma Spotmen. Het ontwikkelen en maken van Spotmen heb ik systematisch aangepakt. Eerst heb ik ervoor gezorgd dat de basiszaken werkten en van daaruit heb ik het programma stap voor stap uitgebreid. Een voorbeeld. Eerst maakte ik het mogelijk dat een QuickTime bestand ingelezen en getoond kan worden in Java, later maakte ik dat mogelijk via een url, daarna bouwde ik de buttons in waarmee je de QuickTime movie kunt besturen, vervolgens werkte ik aan een uitleesbare brontijdcode, en zo verder. Als laatste heb ik het Spotmen programma voorgelegd aan diverse Postmen medewerkers en het commentaar daarop meegenomen in een nieuwe versie. De applicatie werd naarmate mijn Java kennis vorderde steeds verder uitgebreid.

Aangezien ik het voor de klanten van Postmen mogelijk wilde maken om het EDL bestand op te slaan op hun eigen computer, waren de mogelijkheden die een Java applet boden niet voldoende. Java applets zijn beperkt in de mogelijkheden van ‘lokaal bestanden opslaan’ in verband met de veiligheid.

Voor het ontwerpen van het programma zelf heb ik gebruik gemaakt van de *trial and error* methode. Gezien mijn beperkte Java kennis vond ik het niet belangrijk om direct met classes en UML diagrammen te werken, waarmee het programma beter en logischer vormgegeven had kunnen worden. Daardoor werd het programma in eerste instantie bij elke stap groter en ingewikkelder, maar op een gegeven moment bereikte dat weer een omslagpunt en (her-)programmeerde ik alles weer logischer en compacter.

Door veel te programmeren, tussendoor boeken te lezen en van alles te testen, kon de Java code steeds beter toegepast worden op Spotmen. Vanwege de deadline aan het einde van de afstudeerperiode moest ik ook snel keuzes maken over welke Java technieken ik wilde gaan gebruiken. Door met Java en QuickTime for Java te werken en door er verder niet teveel andere Java bij te halen (zoals JDBC voor databases) is het gelukt om een goed werkend prototype van Spotmen te programmeren en te implementeren.

Met het C++ tijdcode programma was duidelijk geworden dat het mogelijk was een tijdcode track uit te lezen. Alleen bleken de C++ functies qua naamgeving en werking af te wijken van de QuickTime for Java classes en ik kwam er niet uit om deze zelf te herschrijven naar Java. Hier heeft de Apple mailinglist me op weg geholpen, want anderhalve week nadat ik er een vraag plaatste, kreeg ik een reactie in de vorm van een bruikbaar stukje code. Het werkte goed en na een aantal aanpassingen kon ik het implementeren in Spotmen.

Rekenen met tijdcodes
Om bepaalde elementen van een EDL bestand te genereren moest er met tijdcodes gerekend worden. Voor het berekenen van bijvoorbeeld de tijdsduur tussen een in en uitpunt tijdcode moesten deze van elkaar worden afgetrokken. Er bestaan speciale tijdcode rekenmachines om dit voor elkaar te krijgen, maar voor Java kon ik geen bestaande class of methode vinden die dit kan doen. Het lastige aan het rekenen met tijdcodes is namelijk dat elk deel (uren, minuten, secondes en frames) weer een eigen maateenheid heeft met een eigen maximale waarde. Stel een sportwedstrijd begint om 13:45:57 en duurt tot 15:23:24 hoe lang heeft deze wedstrijd dan uiteindelijk geduurd? Dit is niet heel gemakkelijk uit te rekenen.

Als eerste probeerde ik de tijdcode String – die uit vier delen bestaat – om te zetten naar de losse variabelen *uren*, *minuten*, *secondes en frames*. Dit werd gedaan met de Java class Tokenizer en als scheidingteken ':' (dubbele punt) gebruikt. De Tokenizer class kan een string opdelen in losse waarden aan de hand van een scheidingsteken. De eerste stap was gelukt, maar aangezien elke variabele dus weer een andere maximale waarde heeft (frames maximaal 25, minuten maximaal 60, etc.) zou het heel wat Java code kosten om dit probleem opgelost te krijgen.

Spotmen genereert de tijdcode string aan de hand van het aantal frames, een variabele die uit één getal bestaat. De tijdcode string 00:00:00:01 kun je ook opschrijven als 1 frame, de tijdcode 00:00:01:14 bestaat uit 39 frames (uitgegaan van 25 frames per seconde). Aangezien het aantal frames uit één getal bestaat valt hier wél gemakkelijk mee te rekenen. Om nu het verschil tussen twee tijdcodes te berekenen – en daarmee het uitpunt van de bestemmingstijdcode – is de werkwijze als volgt. Het aantal frames van het gekozen in-punt kon van het aantal frames van het gekozen uitpunt worden afgetrokken. Op deze manier vond je de *duration (tijdsduur)* van de clip. Vervolgens kon deze waarde bij het inpunt van het vorige inpunt van de bestemmings tijdcode op worden geteld. *Zie de volgende bladzijde voor een voorbeeld.*

Eigenlijk kun je de oplossing die ik heb gevonden vergelijken met het vereenvoudigen van breuken. Als je met breuken rekent wil je dat de noemers van de breuk gelijk zijn, zodat je de tellers eenvoudig kunt op- en aftrekken. De volledige tijdcode waarde zou je kunnen zien als breuk, waarin meerdere noemers zitten: 25 voor het maximumaantal frames, 60 voor de seconden en minuten en 99 voor het aantal uren. Bij het vereenvoudigen zoek je naar de kleinste gemene deler: de frames.

Op de volgende pagina staat een rekenvoorbeeld aan de hand van een EDL bestand.

TITLE:070413_BLACKBOOK
FCM: NON-DROP FRAME

				(Bron in- en uitpunten)		**(Bestemmingsin- en uitpunten)**	
001	*001*	*AA/V*	*C*	*01:00:10:00*	*01:00:11:07*	*00:00:00:00*	*00:00:01:07*
002	*001*	*AA/V*	*C*	*01:00:27:02*	*01:00:28:14*	*00:00:01:07*	*00:00:02:19*
003	*001*	*AA/V*	*C*	*01:00:38:02*	*01:00:39:04*	*00:00:02:19*	*00:00:03:21*

Voor de leesbaarheid is de spatiering aangepast (deze is niet de spatiering van een EDL bestand).
Zie voor uitleg over de functies van de EDL de bijlage EDL breakdown.

De QuickTime movie (een trailer van de film zwartboek) bestaat uit 25 beelden per seconde.
De eerste keer dat er een inpunt wordt gezet wordt het bestemmings in-punt op 00:00:00:00 gezet. Het bestemmings uit-punt is de eerste keer het verschil tussen het in-punt en uit-punt van de brontijdcode, de lengte van dit stuk beelmateriaal dus.

Bij het tweede EDL blokje wordt het bestemmings in-punt gehaald uit het bestemmings uit-punt van het vorige blokje. In dit geval 00:00:01:07.
Het bestemmings uit-punt wordt dan als volgt berekend.
01:00:27:02 wordt als frames opgeslagen, dus 90677 frames.
01:00:28:14 wordt ook als aantal frames (90714 frames) opgeslagen in een variable.
Nu worden deze van elkaar afgetrokken (90714 - 90677 = 37 frames), dit is de tijdsduur van de clip.
Nu wordt het bestemmings in-punt 00:00:01:07 ook in frames omgezet, 32 frames dus.
Vervolgens wordt de tijdsduur van de clip, 37 frames bij de 32 frames opgeteld, dat resulteert in 69 frames.
Dit wordt vervolgens naar tijdcode omgerekend en geeft dan 00:00:02:19.

Vooral deze bovenstaande berekening heeft voor heel wat testen en experimenteren gezorgd om alle variabelen om het juiste punt op te slaan, uit te lezen, of te vervangen.

Overige problemen
Om het EDL bestand te kunnen identificeren, om zo te zien bij welk project of videobestand deze hoort, moest weer een programmeerprobleem opgelost worden. Wat ik voor elkaar wilde krijgen was om de naam van het QuickTime bestand wat de klant gebruikt om te spotten te gebruiken als titel voor het EDL bestand. Dit was bij toeval gelukkig snel op te lossen. De manier was om de naam van een QuickTime movie uit de url te halen. Dit heb ik uiteindelijk voor elkaar gekregen via het volgende stukje code.

```
String movieName = null;
StringTokenizer stringTok = new StringTokenizer(url, "/");

while (stringTok.hasMoreTokens())
{
        movieName = stringTok.nextToken();
}

movieName = movieName.toUpperCase();

return movieName;
```

Deze methode zorgt ervoor dat de url in een string wordt geladen
deze StringTokenizer zoekt vervolgens naar het teken “/” in de url.
Zolang er nog meer / in de url zitten gaat de methode opzoek naar een volgend teken.
Als die niet meer zijn gevonden dan wordt vanaf het laatste / teken de tekst in de string movieName gezet, naar hoofdletters omgezet en terug gegeven.

Op deze manier wordt een url als: *http://www.postmen.tv/spotmen/klantnaam/movie1.mov*
Omgezet naar MOVIE1.MOV

Ten slotte
Door het vele programmeren maakte ik kennis met de voordelen van Java. Een groot voordeel van Java is dat classes en stukken code van diverse bronnen vaak snel in een andere applicatie of applet kunnen worden ingevoegd. Dit was ook het geval bij Spotmen waar ik de printfunctie om de EDL te kunnen afdrukken met de printer snel kon integreren zonder dat ik de code van de printfunctie zelf hoefde te begrijpen.

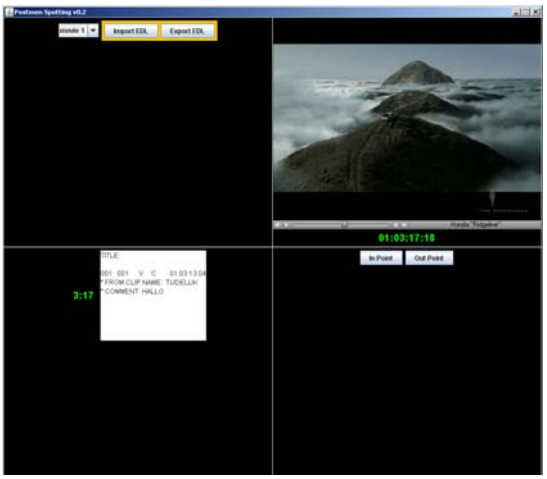
Op de printfunctie na heb ik besloten om de Java code niet op te delen in meerdere classes. Ik heb wel talloze testen uitgevoerd om de layout in een aparte class te stoppen, maar daarbij stuitte ik nog op te veel problemen. Aangezien een werkend programma meer prioriteit had dan een goed ontwerp ben ik hierna niet meer bezig geweest om het programma in classes op te delen.

Toen ik merkte dat ik een stuk code meerdere keren in het programma moest plakken, besloot ik dat dit eenvoudiger kon en heb ik een methode gemaakt. Dit heb ik o.a. toegepast bij het omrekenen van aantal frames naar tijdcode, want dit moest meerdere keren in het programma gebeuren. Hierdoor werd het programma een stuk overzichtelijker en leesbaarder.

Aan het begin werd de code voor Spotmen steeds langer en complexer, maar aan het einde werd het steeds duidelijker en compacter en konden een heleboel dingen die ik had uitgezocht en gecodeerd weer geschrapt worden doordat de oplossing op een andere manier beter was.

Aan het begin van het onderzoek snapte ik nog niet hoe ik met de Java API kon werken, echter na verloop van tijd begreep ik steeds beter het nut ervan en heb ik deze steeds vaker geraadpleegd bij het programmeren. Hierdoor merkte ik duidelijk dat ik met mijn Java kennis vooruit ging.

Alle Java problemen die ik tegen kwam en die relevant waren voor Spotmen heb ik kunnen oplossen.
Als ik er zelf niet uit kwam dan stuurde ik een e-mail met een vraag naar de mailinglist, waar ik dan soms antwoord op kreeg. De docenten die ik benaderde tijdens mijn afstudeerproject konden er niet voor zorgen dat mijn Java problemen werden opgelost. Via talloze testen en debuggen kwam ik er gelukkig zelf uit.



Eerste ontwerpen van Spotmen.

5.4 Server communicatie

Terwijl ik bezig was met het programmeren moest er ook bepaald worden hoe de QuickTime movies, Java programma's en eventueel bijbehorende websites bij de klant terecht konden komen.

Om het Java programma te verspreiden waren er eigenlijk drie mogelijkheden, een Java applicatie, een Java applet en een Java applicatie via Java webstart.

Aangezien het Spotmen programma via internet te benaderen moest zijn was eerst de keus gemaakt voor een Java applet, maar al snel werden de beperkingen duidelijk. Een applet kan bijvoorbeeld niet lokaal bestanden opslaan en heeft nog andere beperkingen, vooral voor de veiligheid van de internetgebruiker. Dit werd naarmate het programma vorderde een steeds grotere beperking, vooral voor het testen, want dan ligt elke beperking in de weg. Via *Applet signing* moest het wel mogelijk zijn om bepaalde restricties op te heffen, maar dat was moeilijker dan verwacht.

De tweede mogelijkheid was een Java applicatie. Een Java applicatie was echter niet zo laagdrempelig voor de klant aangezien er niet simpel een uitvoerbaar programma van Spotmen kan worden gemaakt (geen .exe file). De vraag was dan ook hoe de klant beschikking krijgt tot het programma, dit moest dan via downloaden gebeuren.

Terwijl ik al de keus had gemaakt om het Spotmen programma als applicatie te schrijven ontdekte ik nog een manier van verspreiden, namelijk via Java Web Start.

Wat is Java Web Start
Java Web Start brengt het beste van twee werelden samen: internet en een Java applicatie.
Met Java Web Start is het mogelijk om een applicatie via een website uit te voeren. In tegenstelling tot een Java applet wordt de java applicatie niet in een browser venster uitgevoerd en de veiligheidsbeperkingen zijn minder dan bij een applet. Java Web Start maakt gebruik via een soort XML bestand met de extensie .JNLP (Java Network Launching Protocol). In dit .JNLP bestand staat bijvoorbeeld welke Java class er uitgevoerd moet worden uit welke *package*. Een .JNLP bestand wordt naar de Java Runtime Environment gestuurd, welke vervolgens de Java applicatie download naar de computer van de gebruiker en deze uitvoert.

Belangrijke Java Web Start functies zijn de mogelijkheid om automatisch een Java Runtime Environment te installeren als de gebruiker nog geen Java heeft geïnstalleerd. Als de gebruiker het programma uitvoert wordt altijd gekeken (mits de gebruiker verbonden is met internet) of er al een nieuwere versie is van het programma, zo ja dan wordt deze direct geüpdate zodat de gebruiker altijd de laatste versie heeft.
Java Web Start zit standaard bij de huidige Java versies.

5.5 Testfase

Persoonlijke ervaring met Java Web Start
De techniek van Java Web Start leek mij zeer geschikt voor mijn Spotmen programma, maar voordat alles naar behoren werkte was er al heel wat tijd verstreken. Op het moment dat ik een .JNLP bestand aan de Spotmen applicatie koppelde en deze online op mijn webspace zette, kwam ik tot de ontdekking dat, zodra ik op dit bestand klikte, het werd geopend als een tekstbestand, dus niet via Java Web Start. Na enig onderzoek ontdekte ik dat de webserver speciaal geconfigureerd moet worden om .JNLP bestanden te kunnen interpreteren.

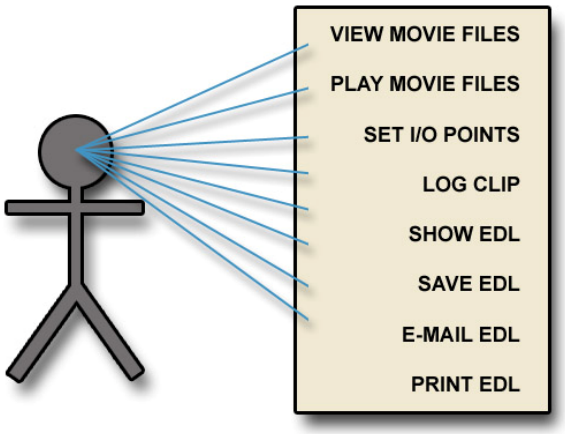
Ik kon nog geen Nederlandse providers vinden die deze dienst ondersteunen dus besloot ik om zelf een Apache Http webserver lokaal bij Postmen te installeren. Vervolgens kon ik dan de volgende regel aan het configuratie bestand van de webserver toevoegen om een nieuwe *MIME-type* toe te voegen: *application/x-java-jnlp-file JNLP* zodat deze wel de .JNLP bestanden goed kon interpreteren.

Een vaste klant van Postmen gebruikt een internetserver waarop hij grote bestanden kwijt kan. Deze server is geschikt voor Spotmen om de QuickTime movies die gespot moeten worden online te kunnen zetten. Nadat ik enige testen had gedaan met deze internet server kwam ik tot de ontdekking dat deze ook de mogelijkheid had om zelf MIME-types toe te voegen zoals .JNLP. Op deze manier kon ik dus de lokale Apache server buiten beschouwing houden, en alles extern via internet draaien.

Terwijl ik bezig was met de webservers en internetserver kwam ik ook tot de ontdekking dat Java Web Start beperkingen heeft. In Spotmen zijn functies ingebouwd om bestanden op te slaan en te importeren. Echter ook de Spotmen applicatie in combinatie met Java Web Start is beveiligd voor zulke functies. De applicatie moet eerst gesigned worden zodat de klant een melding krijgt en kan kiezen om de bestanden lokaal op te slaan, dus de veiligheidsmaatregelen te overschrijden.

De waarschuwing die de klant krijgt als het Spotmen JNLP bestand wordt gestart.

Wensen van de opdrachtgever
Nadat er een redelijk goed werkende versie draaide is het programma getoond aan diverse mensen van Postmen. Zij suggereerden om sneltoetsen in te bouwen waarmee het videobestand op dubbele snelheid afgespeeld kan worden, of waarmee je voor- en achteruit kunt spoelen. Dat zou dan met de J, K en L toets moeten gebeuren, want diezelfde toetsen worden gebruikt in de videomontagesoftware.



Gebruikersinteractie diagram voor Spotmen

Ook suggereerden ze om het plaatsen van in- en uitpunten te verbeteren. In een eerdere versie van Spotmen was de werkwijze van in- en uitpunten zetten als volgt: De klant zette een inpunt en de tijdcode behorend bij dat punt werd dan gelijk getoond in een tekstvlak in Spotmen. Dit tekstvlak toonde dan direct (met een tijdelijke tekst) dat er nog geen bronuitpunt en geen bestemmingsuitpunt was gezet.

Later als dan een uitpunt werd gezet door de klant dan moest alle informatie uit het tekstvlak weer geüpdate worden. Hierbij moesten er strings worden opgezocht en worden vervangen, wat heel wat programmeerwerk en lezen van de Java API met zich mee bracht.

Uiteindelijk wil een klant een inpunt zetten, maar deze is vaak niet definitief dus de mogelijkheid om het inpunt aan te passen moet er wel zijn. Eerst was er ingebouwd dat de klant een melding kreeg dat er eerst een uitpunt gezet moest worden, voordat er nog een keer op inpunt geklikt kon worden. Maar uiteindelijk is het een stuk eenvoudiger geworden. De klant klikt op *in* en er wordt een inpunt gezet, als hij daarna naar een ander tijd in het filmpje gaat en op inpunt drukt, dan wordt de inpunt waarde geüpdate. Dan klikt de klant op uitpunt en wordt er een uitpunt gezet. Pas als de klant daarna op ‘Log Clip’ klikt, wordt het hele blokje toegevoegd aan het tekstvlak. Dit zorgt er voor dat er geen waardes veranderd hoeven worden van het tekstvlak.

Testen in Final Cut Pro
Bij het definitief testen van Final Cut Pro in combinatie met Spotmen stuitte ik nog wel op een aantal problemen. Het commentaar dat je bij de shots kunt geven in de EDL wordt namelijk niet mee geïmporteerd in Final Cut Pro, nu is dit geen grote ramp, aangezien de EDL uitgeprint kan worden met het commentaar.

5.6 Implementatie

Na de research, het programmeren en het eindeloze testen is het tijd om de implementatie fase in te gaan. Dit is het moment om te zorgen dat het programma daadwerkelijk door klanten en door Postmen in gebruik genomen kan worden.

Op dit moment draait het Spotmen programma op een internetserver bij een account van een klant. Het zou verstandig zijn om een Postmen account aan te maken bij http://www.bluehost.com, waarnaar QuickTime movies verstuurd kunnen worden en waar de Spotmen applicatie neer kan worden gezet.

De klanten van Postmen moeten de volgende programmatuur hebben geinstaleerd voordat ze de Spotmen applicatie kunnen gebruiken:

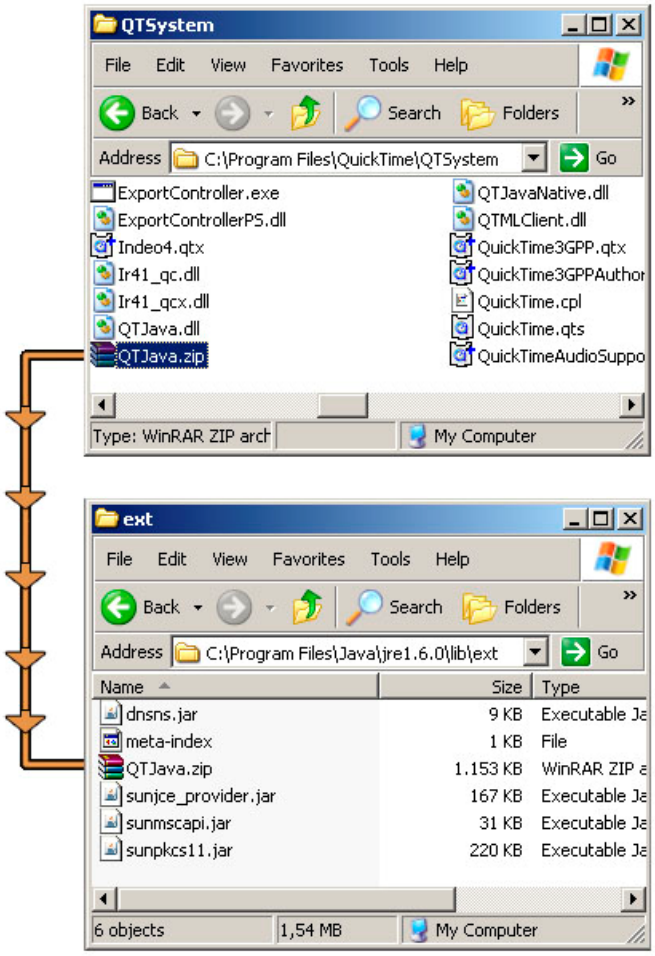
- QuickTime 7 of hoger
- Java Runtime Environment 5 of hoger
- QuickTime for Java library

De klanten met een macintosh computer hebben het gemakkelijk, alle benodigde software is op de mac al geïnstalleerd.

De klanten met een PC hebben het helaas iets minder gemakkelijk. Die klant moet QuickTime en de Java Runtime Environment installeren.

Daarna moet het bestand QTJava.zip uit de submap QTSystem in de map waar QuickTime is geïnstalleerd (C:\Program Files\QuickTime) naar een map van de JRE (C:\Program Files\Java\jre1.6.0\lib\ext) gekopieerd worden.

Dit valt mogelijk via een batch bestand te automatiseren, maar hierbij moet dan wel rekening gehouden worden met het feit dat niet iedereen dezelfde Java versie heeft en niet iedereen de programma's in dezelfde map geïnstalleerd heeft.



Aan de kant van Postmen hoeft niet veel meer te gebeuren. Op dit moment draait Spotmen op een internet server van een klant. Postmen moet een eigen internet server gaan huren, waarop de Spotmen applicatie kan draaien, en waar ook de QuickTime movies op geplaatst kunnen worden. Maar ook zonder die eigen serverruimte kan Postmen op dit moment filmpjes uploaden en daarmee beschikbaar stellen voor de klant.

6. Conclusie en aanbevelingen.

Bij dit afstudeerproject had ik twee doelstellingen: een werkende applicatie afleveren en leren programmeren. Je zou kunnen zeggen dat die twee niet zonder elkaar kunnen en dat is natuurlijk ook zo. Toch wil ik onderstrepen dat ik beide doelstellingen gehaald heb. Een eerste versie van Postmen's spotsysteem *Spotmen* is nu op een externe webserver geïnstalleerd waardoor Spotmen gebruikt kan worden door klanten. Voordat ik aan dit programma begon heb ik mezelf leren programmeren, maar ook tijdens het schrijven van *Spotmen* heb ik nog veel geleerd. Toch valt er in de toekomst nog een hoop te leren over de mogelijkheden van Java en zijn classes.

In deze scriptie ben ik dieper ingegaan op de hoofdvraag en de deelvragen. Concluderend kan ik nu zeggen dat het bedrijfsproces van het spotten geautomatiseerd kan worden met Spotmen. Met dit programma kunnen mensen thuis of op kantoor spotten, waardoor Postmen haar tijd en middelen efficiënter kan indelen.

In het programma zijn de wensen van de opdrachtgever verwerkt. Die wensen zijn geïnventariseerd voordat ik aan het programmeren begon en die zijn later aangescherpt op basis van een eerste versie van Spotmen. Ook de klant heeft zijn mening daarin kunnen geven; Postmen is een erg open bedrijf en dat maakt veel informele ‘opiniepeilingen’ mogelijk.

De laatste versie van Spotmen is nu draaiende, maar op dit moment kan iedereen die de internetlink kent het programma gebruiken. Dat is natuurlijk niet ideaal, het programma zou eigenlijk exclusief voor de klanten van Postmen moeten zijn. Het is ook niet de bedoeling dat klanten het systeem buiten Postmen om gebruiken voor hun eigen materiaal. Momenteel moet de klant een url invullen, wat een url naar elk willekeurig QuickTime bestand kan zijn. Het is wenselijk om daarvoor in de plaats een keuzelijst te maken waaruit de klant filmmateriaal kan kiezen. In deze lijst moet dan alleen het materiaal staan dat Postmen voor de klant online heeft gezet. Idealiter zou er een login systeem voor klanten gemaakt moeten worden, liefst via een MySQL database. Op die manier kunnen klanten makkelijk aan de database toegevoegd worden, waarna ze ook aan een eigen map met videobestanden gekoppeld kunnen worden.

Verder kunnen een aantal zaken in de Java code geoptimaliseerd worden, zoals het toevoegen van elementen aan de EDL. Op dit moment gebeurt dat via tekst 'strings' elementen die toegevoegd worden aan een tekstveld, maar mooier zou het zijn om de EDL als een tabel te zien (via bijvoorbeeld een *Array*). Wellicht zouden deze gegevens aan een MySQL database toegevoegd kunnen worden, zodat spotsessies gemakkelijk gewijzigd, ingevoegd, toegevoegd en weer gewijzigd kunnen worden.

Hieronder is te zien hoe een EDL bestand er in tekstformaat uitziet en hoe deze eruit zou zien als alle informatie netjes in een tabel zou zijn verwerkt. De verschillende elementen hebben allemaal een eigen kleur die correspondeert met de betekenis in de tabel (zie voor uitleg over de EDL elementen bijlage A).

```
001  001      AA/V  C        01:41:55:13 01:42:30:00 00:00:00:00 00:00:34:12
* FROM CLIP NAME:  APOCALYPTO MASTER
AUD   3     4

002  001      AA/V  C        01:42:30:00 01:43:11:16 00:00:34:12 00:01:16:03
* FROM CLIP NAME:  APOCALYPTO MASTER
AUD   3     4
```

Twee gespotte shots in EDL formaat

nr	Clip_nr	r_nr	Aavc	src_in	src_out	Seq_in	seq_out	comment	aud_34
1	001	001	AA/V C	01:41:55:13	01:42:30:00	00:00:00:00	00:00:34:12	INT JUNGLE	AUD 3 4
2	002	001	AA/V C	01:42:30:00	01:43:11:16	00:00:34:12	00:01:16:03	TRACK X	AUD 3 4
3	003	001	AA/V C	01:43:11:16	01:44:28:17	00:01:16:03	00:02:33:04	APE CU	AUD 3 4
4	004	001	AA/V C	01:44:28:17	01:48:34:06	00:02:33:04	00:06:38:18	SPEAR M	AUD 3 4

Vier EDL shots omgezet naar een tabel.

Final Cut Pro heeft de mogelijkheid om een montage te exporteren als een XML bestand. In deze XML staan alle gegevens van het beeldmateriaal en alle instellingen van de *sequence*, zoals de locatie van het gecapturede materiaal, de tijdcodes, audiotracks, compressie, etc. Als in plaats van een EDL bestand een XML bestand gebruikt zou worden, zou er nog meer geautomatiseerd kunnen worden. De werkwijze wordt dan ook enigszins aangepast: Het originele beeldmateriaal wordt gecaptured en het Final Cut Pro project wordt geëxporteerd naar XML bestand. Dit bestand wordt dan samen met het videomateriaal op een server gezet. De klant laadt het XML bestand in de Spotmen applicatie, waarna de klant het XML bestand kan aanpassen bij het spotten. Vervolgens stuurt de klant het geüpdate bestand terug naar Postmen, waar het weer in Final Cut Pro ingeladen kan worden.

Op dit moment is het mogelijk om een EDL bestand als tekst in de body van een email te verzenden. Het zou echter efficiënter en handiger zijn als het bestand als bijlage meegestuurd zou kunnen worden.

Deze eerste versie van Spotmen is geschreven voor het Europese PAL videoformaat. Dit materiaal heeft altijd 25 frames per seconde. In de toekomst kan wellicht gekeken worden of het ook mogelijk is om met het NTSC formaat te werken, waarin 29,97 of 30 frames in een seconde worden afgespeeld.

Tenslotte zou het uitermate handig zijn om ook het proces vanuit Final Cut Pro te automatiseren. Bijvoorbeeld zodat Final Cut Pro na het inladen van de tape automatisch een lage resolutie proxy bestand aanmaakt en dat vervolgens upload. De laatste versie van Final Cut Pro is helaas nog niet *scriptable*, maar het zou mooi zijn als er een programma of script geschreven kan worden dat een bepaalde map in de gaten houdt en kijkt of er nieuwe bestanden in staan. Mocht in die map een hoge resolutie QuickTime bestand terecht komen, dan zou dat automatisch omgezet moeten worden naar een lage resolutie bestand. Vervolgens wordt dat op een server gezet en ontvangt de klant een email met de link naar het programma of het bestand. Hierin staat dan dat het door de klant aangeleverde filmmateriaal klaar staat om gespot te worden.

Screenshot van Spotmen 'Log Clip' mode

7. Evaluatie.

Nog een kleine maand te gaan en ik ben klaar met mijn afstuderen. Als ik terugkijk op de afgelopen periode ben ik erg tevreden over de behaalde resultaten. Mijn programmeerkennis is aanzienlijk verbeterd. Ik heb gemerkt dat als je een tijd niets doet met die kennis, dat alles snel weer wegzakt, omdat ik de laatste jaren van de opleiding eigenlijk alleen maar bezig ben geweest met andere aspecten van de mediatechnologie.

Afstuderen met een programmeeropdracht leek me echt een uitdaging, een soort bekroning op al die jaren mediatechnologie. Het feit dat ik deze opdracht in een postproductiehuis moest vervaardigen gaf mijn motivatie een extra stimulans. Mijn interesse is op school (maar ook in mijn vrije tijd) altijd uitgegaan naar die postproductiekant van de opleiding.

Eenmaal aan de opdracht begonnen kwam ik al vrij snel op de combinatie QuickTime en Java terecht, dat maakte de applicatie immers ‘vanzelf’ crossplatform. Het leren van Java viel me in eerste instantie tegen, er zijn dan ook heel veel verschillende manieren waarop je een programma kunt programmeren. De code in het boek over QuickTime voor Java vond ik niet de makkelijkste, maar na talloze *copy en paste* sessies, kon ik er wel mee overweg. Verder is er nog heel veel mogelijk waar ik me achteraf gezien nog best in had willen verdiepen, bijvoorbeeld in het maken van databases. Wellicht had ik mijn programma dan nog beter kunnen maken.

Op dit moment beheers ik Java op een niveau dat ik redelijk snel fouten kan ontdekken en zaken kan aanpassen. *Events* maken, *if loopjes*, componenten op het scherm zetten, al dan niet met *layoutmanagers*: het gaat allemaal relatief snel. Het Java classes systeem begrijp ik nog niet helemaal. Dat is ook niet heel erg, de prioriteit lag bij het programmeren vooral bij de functionaliteit en niet bij de structuur van de programmeertaal. Tijdens eerdere projecten van de opleiding mediatechnologie werd juist veel nadruk gelegd op het ontwerp van het programmeren, waardoor de functionaliteit vaak te wensen overliet. Dat is jammer, want daardoor zijn het bij mij vaak een beetje net-niet projecten. Dat geldt bijvoorbeeld voor het project dat ik met mijn eigen werkgroep deed: het ontwerpen van een flash netwerk game.

Als ik het programma later nog ga uitbreiden en ik tijd heb om de Javaboeken in te duiken dan doe ik dat graag.

Middels deze scriptie heb ik laten zien hoe er in relatief korte tijd (twintig weken) een prototype van een spotsysteem kan worden gemaakt. De Postmen workflow voor het spotten is hiermee aanmerkelijk verbeterd.

Ik heb gemerkt dat ik goed kon meepraten binnen het bedrijf en dat de termen uit de wereld van de postproductie mij al redelijk bekend waren. Dat komt door de kennis die ik op school heb meegekregen – vooral bij het semester Broadcast & Crossmedia Technology. Maar ook door mijn stage bij Hithouse studio's was ik al in aanraking gekomen met de mediawereld en met de termen die er gebezigd worden. Afsluitend zou je kunnen zeggen dat de opleiding mij van een goede basis heeft voorzien. Goed genoeg om een baan bij Postmen aangeboden te krijgen.

Bronnen.

Geraadpleegde literatuur:

QuickTime for Java - A Developer's Notebook - Chris Adamson, 2005, eerste editie

Java voor studenten – Douglas Bell, Mike Parr, 2003, derde editie

SAMS Teach Yourself Programming with Java in 24 Hours - Rogers Cadenhead, 2005, vierde editie

Java How To Program – Deitel, 2004, zesde editie

Creating Web Applets With Java – David Gulbransen, Kenrick Rawlings, 1996

Black Art of Java Game Programming – Joel Fan, Eric Ries, Calin Tenitchi, 1996

SAMS Teach Yourself Java in 21 Days – Rogers Cadenhead, 2004, vierde editie

Developing Professional Java Applets – K.C. Hopson, Stephen E. Ingram, 1996

Introduction to Computing and Programming with Java: A Multimedia Approach –
Mark Guzdial, Barbara Ericson, 2006

Weblinks:

http://www.apple.com
Algemene website van Apple waar links op staan naar QuickTime.

http://developer.apple.com
De website voor ontwikkelaars. Via deze website kom je bij de onder andere bij de developers informatie voor QuickTime en Java.

http://developer.apple.com/quicktime/qtjava/
De website over QuickTime for Java van Apple.

http://macslash.org/comments.pl?sid=4993&op=&threshold=0&commentsort=0&mode=thread&pid=0
Op deze website staat een link naar een programma om tijdcode aan een QuickTime movie toe te voegen.

http://java.sun.com/docs/white/langenv/index.html
White paper over de Java programmeertaal

http://java.sun.com/docs/books/tutorial/index.html
Algemene Java tutorial

http://www.onjava.com/pub/a/onjava/2003/05/14/qtj_reintro.html
Een artikel over QuickTime for Java, geschreven door Chris Adamson, de auteur van het boek QuickTime for Java A Developer's Notebook.

http://www.edlmax.com/maxguide.html
Website met technische informatie over het EDL formaat.

Bijlagen

Bijlage A: EDL Breakdown

Bijlage B: Originele planning

Bijlage A: EDL Breakdown

Een QuicktTime movie geëxporteerd (in Cmx3600 formaat) uit Final Cut Pro als EDL kan er als volgt uitzien:

```
TITLE: 070412_APOCALYPTO
FCM: NON-DROP FRAME

001  001      AA/V  C        01:41:55:13 01:42:30:00 00:00:00:00 00:00:34:12
* FROM CLIP NAME:  APOCALYPTO MASTER
* COMMENT: CU MAYA LEADER
AUD   3     4

002  001      AA/V  C        01:42:30:00 01:43:11:16 00:00:34:12 00:01:16:03
* FROM CLIP NAME:  APOCALYPTO MASTER
AUD   3     4

003  001      AA/V  C        01:43:11:16 01:44:28:17 00:01:16:03 00:02:33:04
* FROM CLIP NAME:  APOCALYPTO MASTER
AUD   3     4

004  001      AA/V  C        01:44:28:17 01:48:34:06 00:02:33:04 00:06:38:18
* FROM CLIP NAME:  APOCALYPTO MASTER
AUD   3     4
```

Een EDL bestand (in Cmx3600 formaat) kan uit de volgende elementen bestaan. De spatiering tussen de verschillende elementen dient ook meegenomen te worden.

(dit is maar een kleine greep van de elementen van een EDL. Alleen diegene die in Spotmen komen zijn hier beschreven, bij de bronvermelding staat nog een link naar meer mogelijkheden van een EDL).

TITLE: 070412_APOCALIPTO achter TITLE volgt de naam van de sequence.

FCM: NON-DROP FRAME na FCM volgt het frame tijdcode formaat, voor Spotmen heb ik besloten om deze eerst voor alleen PAL geschikt te maken, deze bestaat altijd uit non-drop frame tijdcode.

001 Dit eerste nummer is het clipnummer, in bovenstaand voorbeeld bestaat de montage uiteindelijk dus uit een totaal van vier clips.

001 Het tweede nummer staat voor de band. Als een montage gebruik maakt van meerdere banden, dan loopt dit nummer dus op. Het is ook mogelijk om de band in plaats van een nummer een naam te geven van maximaal acht letters en/of cijfers.

AA/V De eerste twee tekens AA staan voor A1 en A2 oftewel Audio1 en Audio2 kanalen. Meestal staat dit voor een stereo track op de band (A1 linker kanaal, A2 rechter kanaal), maar het is ook mogelijk dat A1 en A2 beide van andere bronnen komen. De V staat voor een videokanaal.

C De ‘C’ staat voor *cut*. Om een *cut* te maken met een EDL moet deze uit twee EDL blokjes bestaan, waarbij de eerste het element ‘C’ moet bevatten. Naast cuts zijn er ook andere overgangen mogelijk.

01:41:55:13 Dit eerste element met tijdcode is het inpunt van de tijdcode van het bronmateriaal.

01:42:30:00 Dit tweede element met tijdcode is het uitpunt van de tijdcode van het bronmateriaal.

00:00:00:00 Dit derde element is het inpunt op de tijdlijn van het montageprogramma (de bestemmingstijdcode). In Spotmen is het zo ingesteld dat deze altijd op 00:00:00:00 begint, maar vaak wordt ook 01:00:00:00 gebruikt als beginwaarde.

00:00:34:12 Dit vierde element is het uitpunt van de bestemmingstijdcode (tijdlijn van het montageprogramma).

* FROM CLIP NAME: APOCALYPTO MASTER Hier staat de naam van de masterclip.
* COMMENT: CU MAYA LEADER Hier kan commentaar worden toegevoegd aan de clips.

AUD 3 4 Dit element wil zeggen dat de montage ook een audiokanaal drie en vier bevat (in totaal vier).

Bijlage B: Originele planning:

Weeknr.	Datum	Geplande werkzaamheden	
49	4 dec.	VOORBEREIDEN, programmeertaal bepalen	
50	11 dec.	FCP uitzoeken, proxyworkflow, Qtcomp bepalen, edl uitzoeken(evt ook met Avid).	
51	18 dec.		
52	25 dec.	vakantie	
1	01 jan.	webomgeving bouwen filmpjes, in/out points, edl genereren en commentaar invoeren.	
2	08 jan.		
3	15 jan.		
4	22 jan.		
5	29 jan.		
6	05 feb.	server uitzoeken wat het bestevoordeligste is: intern/extern	
7	12 feb.	webomgeving bouwen accounts	
8	19 feb.		
9	26 feb.		
10	05 mrt.		
11	12 mrt.		
12	19 mrt	Server implementatie	
13	26 mrt.		
14	02 apr.	Afronden en testen	
15	09 apr.		
16	16 apr.	verslag	
17	23 apr.	verslag	

Deze planning is aan het begin van de afstudeerperiode gemaakt, echter geeft deze uiteindelijk niet een realistisch beeld weer van de werkzaamheden die daadwerkelijk uitgevoerd zijn. Dit valt te wijten aan ongeplande werkzaamheden en zaken die langer duurden dan gepland of kwamen te vervalllen.